Inconsistency Handling in Ontology-Mediated Query Answering

Camille Bourgaux

NNT : 2016SACLS292

Thèse de doctorat de l'Université Paris-Saclay préparée à l'Université Paris-Sud

Ecole doctorale n°580
Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Informatique

par

Mme Camille Bourgaux

Gestion des incohérences pour l'accès aux données en présence d'ontologies

Thèse présentée et soutenue à Orsay, le 29 septembre 2016.

Composition du Jury :

Mme	Christine Froidevaux	Professeur Université Paris-Sud	(Présidente)
M.	Maurizio Lenzerini	Professeur Sapienza Università di Roma	(Rapporteur)
Mme	Marie-Laure Mugnier	Professeur Université Montpellier	(Rapporteur)
Mme	Marie-Christine Rousset	Professeur Université de Grenoble	(Examinatrice)
Mme	Meghyn Bienvenu	Chargée de recherche CNRS, Université Montpellier	(Directrice de thèse)
M.	François Goasdoué	Professeur Université Rennes 1	(Directeur de thèse)

REMERCIEMENTS

Je tiens à remercier tous ceux qui ont permis que ce travail de thèse se déroule dans de bonnes conditions.

En premier lieu, un très grand merci à Meghyn et François pour avoir été d'excellents directeurs de thèse. Merci pour votre confiance et votre disponibilité. Vos conseils et votre aide ont été précieux. Je suis consciente de la chance que j'ai eu de travailler avec vous et espère en avoir encore l'occasion.

Je remercie vivement Maurizio Lenzerini et Marie-Laure Mugnier d'avoir accepté d'être rapporteurs et pour leur relecture attentive de ce manuscrit. Merci également à Christine Froidevaux et Marie-Christine Rousset d'avoir accepté de faire partie de mon jury.

Merci aussi à tous ceux que j'ai rencontrés dans le cadre de cette thèse: mes collègues du LRI, membres de l'équipe LaHDAK, des équipes pédagogiques dans lesquelles j'ai effectué mon enseignement, ou des équipes administratives et techniques; ceux de l'équipe Inria CEDAR, en particulier ceux qui m'ont aidée pour les expérimentations, et ceux de l'équipe GraphIK du LIRMM qui m'a accueillie pendant trois mois. Je remercie en particulier les membres du projet PAGODA pour les discussions enrichissantes que nous avons eues.

Enfin, merci à mes proches qui m'ont accompagnée durant ces trois ans. Merci pour votre soutien et les bons moments partagés.

ABSTRACT

The problem of querying description logic knowledge bases using database-style queries (in particular, conjunctive queries) has been a major focus of recent description logic research. An important issue that arises in this context is how to handle the case in which the data is inconsistent with the ontology. Indeed, since in classical logic an inconsistent logical theory implies every formula, inconsistency-tolerant semantics are needed to obtain meaningful answers. This thesis aims to develop methods for dealing with inconsistent description logic knowledge bases using three natural semantics (AR, IAR, and brave) previously proposed in the literature and that rely on the notion of a repair, which is an inclusion-maximal subset of the data consistent with the ontology. In our framework, these three semantics are used conjointly to identify answers with different levels of confidence. In addition to developing efficient algorithms for query answering over inconsistent DL-Lite knowledge bases, we address three problems that should support the adoption of this framework: (i) query result explanation, to help the user to understand why a given answer was (not) obtained under one of the three semantics, (ii) query-driven repairing, to exploit user feedback about errors or omissions in the query results to improve the data quality, and (iii) preferred repair semantics, to take into account the reliability of the data. For each of these three topics, we developed a formal framework, analyzed the complexity of the relevant reasoning problems, and proposed and implemented algorithms, which we empirically studied over an inconsistent DL-Lite benchmark we built. Our results indicate that even if the problems related to dealing with inconsistent DL-Lite knowledge bases are theoretically hard, they can often be solved efficiently in practice by using tractable approximations and features of modern SAT solvers.

Résumé

Interroger des bases de connaissances avec des requêtes conjonctives a été une préoccupation majeure de la recherche récente en logique de description. Une question importante qui se pose dans ce contexte est la gestion de données incohérentes avec l'ontologie. En effet, une théorie logique incohérente impliquant toute formule sous la sémantique classique, l'utilisation de sémantiques tolérantes aux incohérences est nécessaire pour obtenir des réponses pertinentes. Le but de cette thèse est de développer des méthodes pour gérer des bases de connaissances incohérentes en utilisant trois sémantiques naturelles (AR, IAR et brave) proposées dans la littérature et qui reposent sur la notion de réparation, définie comme un sous-ensemble maximal des données cohérent avec l'ontologie. Nous utilisons ces trois sémantiques conjointement pour identifier les réponses associées à différents niveaux de confiance. En plus de développer des algorithmes efficaces pour interroger des bases de connaissances DL-Lite incohérentes, nous abordons trois problèmes: (i) l'explication des résultats des requêtes, pour aider l'utilisateur à comprendre pourquoi une réponse est (ou n'est pas) obtenue sous une des trois sémantiques, (ii) la réparation des données guidée par les requêtes, pour améliorer la qualité des données en capitalisant sur les retours des utilisateurs sur les résultats de la requête, et (iii) la définition de variantes des sémantiques à l'aide de réparations préférées pour prendre en compte la fiabilité des données. Pour chacune de ces trois questions, nous développons un cadre formel, analysons la complexité des problèmes de raisonnement associés, et proposons et mettons en oeuvre des algorithmes, qui sont étudiés empiriquement sur un jeu de bases de connaissance DL-Lite incohérentes que nous avons construit. Nos résultats indiquent que même si les problèmes à traiter sont théoriquement durs, ils peuvent souvent être résolus efficacement dans la pratique en utilisant des approximations et des fonctionnalités des SAT solveurs modernes.

Table of contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

1

INTRODUCTION

Ontology-mediated query answering

With the increase of the volume of available data comes the problem of being able to exploit it. Querying data in an accurate and efficient way is a complex task and has attracted a lot of interest. Among the challenges to address in this context, it is necessary to allow the integration of data coming from different sources using different vocabularies and the formulation of queries in a simple and intuitive way, with a vocabulary that closely matches the user's conceptualization of the world. For instance, suppose that someone needs to find the professors of a university department who teach a course related to the topic of artificial intelligence. Suppose that he has got a record of the department members with their positions, and a list of courses with their instructors. If he simply integrates this data into a traditional database and queries it in the straightforward way, asking "select every professor who teaches artificial intelligence", he may encounter two problems: first, the positions of the department members will not be recorded simply as "professor" but rather with their degree of seniority ("full professor", "associate professor" and "assistant professor" for instance), second many courses that concern a subfield of artificial intelligence will be missed by such a query (for instance courses recorded as "machine learning", "knowledge representation", or "description logics" will not be recognized as artificial intelligence courses). He therefore has to find first the different denominations that correspond to a professor position and the different fields and subfields of artificial intelligence, and reformulate his query accordingly, which may become quite complex. Ontology-mediated query answering is a recent paradigm that adds a semantic layer on top of the data by the means of a logical theory called an ontology, which formalizes knowledge about a particular domain and is used to reason about the data in order to provide more complete answers to queries. In our example, adding to the dataset an ontology that provides knowledge about the fields and subfields of computer science as well as about the university organization will allow the user to get every relevant answer while posing his question in the way which is natural for him.

Description logics [Baader *et al.* 2003] are a family of fragments of classical first-order logics which are widely used as ontology languages. A description logic knowledge base consists in an ontology, called a TBox, which expresses general knowledge and rules about the domain of interest, and a dataset, called an ABox, which provides information about specific individuals. For instance, the TBox of a knowledge base about the university domain could indicate that courses are taught by faculty members and followed by students, and its ABox could state that an individual named Ann gives a database course which is taken by another individual Bob.

Enriching the data with an ontology comes at a price: using an ontology increases the computational complexity of query answering. Since scalability is essential for data-rich applications, there has been an increasing interest in lightweight description logics, which offer a good trade-off between the expressivity of the language and the computational complexity of the associated reasoning problems. In particular, the DL-Lite family [Calvanese *et al.* 2007] has been especially tailored for ontology-mediated query answering and allows it to be reduced, via query rewriting, to standard database query evaluation. In this thesis we adopt the language DL-Lite$_{\mathcal{R}}$, which is the dialect of the DL-Lite family that underlies OWL 2 QL [Motik *et al.* 2012], i.e. the OWL2 profile devoted to query answering of the standard for the Semantic Web.

Inconsistency handling

An important issue that arises in the context of ontology-mediated query answering is how to handle the case in which the data is inconsistent with the ontology. Indeed, while the TBox is generally rather small and extensively debugged by experts of the domain, the ABox is typically large, subject to frequent modifications, and may results from the integration of different data sources, which make errors likely. The problem is that an inconsistent knowledge base implies any logical sentence, so posing a query over such a knowledge base will result in getting every possible answer formed with the individuals of the knowledge base as an answer. For instance, if a knowledge base indicates that it is not possible to be a full professor and an assistant professor at the same time, and that an individual Ann is said to be both, the knowledge base will allow us to derive not only that Ann is a full and an assistant professor, but also that Ann is a course for example, which is clearly undesirable.

There are two possible attitudes in this context. The first one is to restore the consistency, for instance by dropping one or both assertions about the seniority of Ann in our example, but it may be impossible to do that in a satisfying way. Indeed, we often do not know how to repair the data (is Ann an assistant, or a full professor?), and removing every piece of information that is doubtful because involved in some contradiction will often lead to an unacceptable loss of information. Moreover, examining manually every conflict in the data will be too costly when repairing a large dataset. The second option is to decide to live with the inconsistencies, by trying to get meaningful answers to queries from inconsistent knowledge bases. For instance, it may be acceptable to derive that Ann is a professor, but not that she is a course. Several inconsistency-tolerant semantics have been defined to achieve this goal. The most well-known is the AR semantics [Lembo *et al.* 2010], which retrieves the answers which hold in each of the maximal consistent subsets of the data, called repairs. This semantics amounts to considering true the answers that hold no matter which possible world is chosen. For instance, it will allow us to find that Ann is a professor, without any information on her degree of seniority. The drawback of this semantics is that it is computationally hard. Indeed, over DL-Lite knowledge bases, conjunctive query entailment under AR semantics is intractable, even when the complexity is measured only w.r.t. the size of the data. To overcome this difficulty, a data-tractable sound approximation of AR, called the IAR semantics, has been introduced in [Lembo *et al.* 2010]. The IAR-answers are obtained by querying the intersection of the repairs. This semantics is also interesting by itself, because it retains only the "surest" answers, whose supports are not involved in any contradiction. Under this semantics, no information about Ann will be retrieved, because the reasons for thinking that Ann is a professor are both contradicted, so not completely reliable. At the other end of the scale, the brave semantics provides every answer that holds in some repair [Bienvenu & Rosati 2013]. It may indeed be important for some applications not to miss any possible answer that has some consistent reason to hold. Using the brave semantics allows us to find that Ann may be a full professor as well as she may be an assistant professor.

Contributions

This thesis aims at developing methods to practically deal with inconsistent knowledge bases. In particular, we defend the idea that even if the AR semantics is intractable, it can be used in practice.

Our first contribution is indeed an approach for classifying the answers into classes of increasing reliability, depending on they are entailed under IAR, AR or brave semantics. For the AR semantics, the brave and IAR semantics provide tractable upper and lower bounds and a translation into propositional satisfiability allows us to decide whether the brave and non-IAR answers are entailed under AR.

Beyond efficient query answering, it is important to be able to provide explanations of the query results under inconsistency-tolerant query answering. Indeed, a user may naturally wonder why an answer belongs to one of these classes (e.g. why Ann is said to be a professor under AR semantics but not under IAR semantics?). That is why our second contribution is a framework for explaining positive and negative answers of a query under AR, IAR or brave semantics (e.g. Ann is probably a professor because she is either an assistant professor or a full professor in every possible world, but none of these reasons is beyond doubt since they contradict each other). We believe that such facilities are essential to make inconsistency-tolerant semantics really usable.

Our third contribution is a query-driven approach for partially repairing the data. Indeed, while alternative semantics are necessary to deal with inconsistent knowledge bases, they cannot replace the need for improving the data quality. We propose to exploit the feedback of the user at query time about the answers that are correct or incorrect to clean the dataset, focusing on the part which is useful for the user needs and that he knows well enough to repair (e.g. if the user knows that Ann is actually an assistant professor, we can delete the data asserting that she is a full professor, since we know that she cannot be both).

The last contribution of the thesis is the investigation of variants of AR, IAR and brave semantics obtained by replacing the classical notion of a repair by preferred repairs. This allows us to take into account the information about the reliability of the data (e.g. if the piece of information that Ann is an assistant professor comes from a more reliable source than those that Ann is a full professor, we can keep only the repairs that contain the former, and conclude that Ann is a professor even under IAR semantics).

For each of the issues we consider, we analyze the computational complexity of the related problems, and propose algorithms to solve them, relying on performances of modern-day SAT solvers to practically handle those which are hard. We implemented most of these algorithms in our CQAPri[1] prototype and empirically study their properties using a benchmark we built upon the well-known DL-Lite$_{\mathcal{R}}$ benchmark LUBM$^{\exists}_{20}$.

Organization of the thesis

The thesis is organized as follows:

Chapter 2 This chapter introduces the framework of ontology-mediated query answering in the setting of description logics, focusing on the lightweight description logic DL-Lite$_{\mathcal{R}}$ that we adopt in this work. In the second half of chapter, we review the alternative semantics that have been proposed to deal with inconsistent data in this context.

Chapter 3 In this chapter, we present the algorithms implemented in our CQAPri prototype system for query answering under AR, IAR and brave semantics over DL-Lite$_{\mathcal{R}}$ knowledge bases. Then, we describe the experimental setting we built to evaluate it, as well as the results we obtained, and briefly discuss other existing systems and benchmarks for inconsistency-tolerant query answering.

Chapter 4 We address in this chapter the problem of explaining why a tuple is or is not an answer to a query under the IAR, AR or brave semantics. We define data-centric explanations for positive and negative answers, study their computational complexity in DL-Lite$_{\mathcal{R}}$, and propose algorithms to compute them by exploiting solvers for Boolean satisfaction and optimization problems. We also present our implementation within CQAPri and the experiments we conducted.

Chapter 5 This chapter addresses the problem of query-driven repairing of inconsistent DL-Lite$_{\mathcal{R}}$ knowledge bases. The scenario we consider is the following: a user receives query answers under inconsistency-tolerant semantics, and provides feedback about which answers are erroneous, or are indeed answers and should hold under a stronger semantics. The aim is to find a set of ABox modifications (deletions and additions), called a repair plan, that addresses as many of the defects as possible. After formalizing this problem and introducing different notions of optimality, we investigate the computational complexity of reasoning about optimal repair plans and propose interactive algorithms for computing such plans. For deletion-only repair plans, we propose an improved algorithm and present the implementation of its core components in our CQAPri system.

Chapter 6 In this chapter, we investigate variants of the AR, IAR and brave semantics obtained by replacing the classical notion of repair by one of four different types of preferred

[1]available at www.lri.fr/~bourgaux/CQAPri

repairs (e.g. cardinality-maximal repairs, or repairs based on priority levels that distinguish more or less reliable assertions). We analyze the complexity of query answering under the resulting semantics, and propose an approach exploiting a SAT encoding for the semantics based on repairs using priority levels, whose data complexity is "only" coNP-complete, as for plain AR semantics. We then present our implementation of these semantics and its experimental evaluation.

Chapter 7 In this chapter, we position our work in a more general context and provide more details on some topics mentioned in the other chapters.

Chapter 8 This chapter summarizes our contributions and indicates some possible extensions of this work.

Appendix A The appendix provides basics elements of computational complexity theory and of propositional logic, and recalls the definitions of the different complexity classes appearing in this thesis and of the problems used in the complexity proofs.

2

PRELIMINARIES

In this chapter, we first introduce the framework of ontology-mediated query answering in the setting of description logics. We focus in this thesis on the lightweight description logic DL-Lite that has been especially designed for scalable query answering, and in particular on its dialect DL-Lite$_\mathcal{R}$. In the second half of chapter, we review the alternative semantics that have been proposed to deal with inconsistent data in the context of ontology-mediated query answering.

2.1 Ontology-mediated query answering in DL-Lite

Ontologies are logical theories that formalize the knowledge about a domain of interest. Using such formalism makes information sharing easier by providing a standardized terminology with an unambiguous semantics. Moreover, by making knowledge processable by computers, ontologies support complex automatic reasoning to infer new knowledge from that originally formulated by some experts of the domain. Recent years have seen an increasing interest in *ontology-mediated query answering* (OMQA) in which conceptual knowledge provided by an ontology is exploited when querying incomplete data (cf. [Bienvenu & Ortiz 2015] for a survey). The ontology provides an enriched vocabulary that may help users to formulate queries in a terminology that is closer to the one they are familiar with, without having to take into account the way the data is stored and the vocabulary of the database schema. It also allows the integration of different data sources using different vocabularies in one single conceptual model. Moreover, OMQA provides more complete answers to queries by taking into account implicit consequences of the facts stored in the data that follow from the rules formalized in the ontology.

2.1.1 Description logic basics

Description logics (DLs) [Baader *et al.* 2003] are a family of decidable fragments of classical first-order predicate logic, which are popular ontology languages. They provide the basis for the web ontology language (OWL) [Motik *et al.* 2012], which is the standard of the World Wide Web Consortium (W3C) for the Semantic Web.

Syntax

A DL *knowledge base* is built using a DL *vocabulary* consisting of three countably infinite, pairwise disjoint sets of symbols:

- $\mathsf{N_I}$ is the set of *individuals*, which are constants referring to specific objects of the domain of interest
- $\mathsf{N_C}$ is the set of *concept* names, which are unary predicates representing sets of objects sharing some common characteristics. Concepts are called *classes* in OWL terminology.
- $\mathsf{N_R}$ is the set of *role* names, which are binary predicates representing relationships between objects. Roles are OWL *properties*.

In this thesis, we will use the university domain for examples.

Example 2.1.1. To build a knowledge base about the university domain, we need to refer to specific university employees, students, courses, and departments like ann, bob, $carl$, $DBcourse$, $AIcourse$, $CSdepartment$: they are individuals of $\mathsf{N_I}$. To represent sets of individuals like professors, students, courses, or departments, we need concept names: Prof, $\mathsf{Student}$, Course, $\mathsf{Department}$ are in $\mathsf{N_C}$. To talk about relations between them, for instance to express that someone teaches a course, advises a student, or is member of a department, we use role names: Teach, Advise, $\mathsf{MemberOf}$ are in $\mathsf{N_R}$. $\triangleleft$

This vocabulary is used to represent the knowledge about the domain of interest with two kinds of axioms. *Terminological axioms* express properties of concepts and roles and relations between them, thus specify intensional knowledge, that concern whole groups of individuals (e.g. all professors are PhD holders, what is taught is a course). *Assertions* are facts about specific individuals which express that an individual belongs to some concept or that a role links two given individuals (e.g. Ann is a professor, Ann teaches the database course), thus specify extensional knowledge.

Definition 2.1.2 (Knowledge base)**.** A *TBox* is a finite set of terminological axioms and an *ABox* is a finite set of assertions. A *knowledge base* (KB) $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is composed of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. The set of individuals that appear in assertions of $\mathcal{A}$ is denoted by $\mathsf{Ind}(\mathcal{A})$.

Terminological axioms can involve *complex concepts and roles* built from the symbols in the vocabulary using *constructors*. Each particular DL is distinguished by the constructors it offers and restrictions on how to use them in terminological axioms, that allow more or less expressivity. Tables 2.1 and 2.2 show common constructors.

Example 2.1.3. Assertions can state that Ann is a professor ($\mathsf{Prof}(ann)$) that teaches the database course ($\mathsf{Teach}(ann, DBcourse)$) and advises Bob ($\mathsf{Advise}(ann, bob)$).

Terminological axioms can express that every professor is a member of some department ($\mathsf{Prof} \sqsubseteq \exists \mathsf{MemberOf}.\mathsf{Department}$), that what is taught is a course ($\exists \mathsf{Teach}^- \sqsubseteq \mathsf{Course}$), or that every professor is either a full professor or an associate professor ($\mathsf{Prof} \sqsubseteq \mathsf{FProf} \sqcup \mathsf{AProf}$), but that a full professor cannot be also an associate professor ($\mathsf{FProf} \sqsubseteq \neg \mathsf{AProf}$). $\triangleleft$

Table 2.1 Syntax and semantics of DL concept constructors: a denotes an individual name, C, C_1 and C_2 (complex) concepts, R a (complex) role, and n a natural number.

Name	Syntax	Semantics
Top concept	$\top$	$\Delta^{\mathcal{I}}$
Bottom concept	$\bot$	$\emptyset$
Nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
Negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Conjunction	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
Disjunction	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
Existential restriction	$\exists R$	$\{d_1 \mid$ there exists $(d_1,d_2) \in R^{\mathcal{I}}\}$
Qualified existential restriction	$\exists R.C$	$\{d_1 \mid$ there exists $(d_1,d_2) \in R^{\mathcal{I}}$ with $d_2 \in C^{\mathcal{I}}\}$
Qualified universal restriction	$\forall R.C$	$\{d_1 \mid d_2 \in C^{\mathcal{I}}$ for all $(d_1,d_2) \in R^{\mathcal{I}}\}$
Unqualified number restrictions	$\geq nR$	$\{d_1 \mid \lvert\{d_2 \mid (d_1,d_2) \in R^{\mathcal{I}}\}\rvert \geq n\}$
	$\leq nR$	$\{d_1 \mid \lvert\{d_2 \mid (d_1,d_2) \in R^{\mathcal{I}}\}\rvert \leq n\}$
	$= nR$	$\{d_1 \mid \lvert\{d_2 \mid (d_1,d_2) \in R^{\mathcal{I}}\}\rvert = n\}$
Qualified number restrictions	$\geq nR.C$	$\{d_1 \mid \lvert\{d_2 \mid (d_1,d_2) \in R^{\mathcal{I}}$ and $d_2 \in C^{\mathcal{I}}\}\rvert \geq n\}$
	$\leq nR.C$	$\{d_1 \mid \lvert\{d_2 \mid (d_1,d_2) \in R^{\mathcal{I}}$ and $d_2 \in C^{\mathcal{I}}\}\rvert \leq n\}$
	$= nR.C$	$\{d_1 \mid \lvert\{d_2 \mid (d_1,d_2) \in R^{\mathcal{I}}$ and $d_2 \in C^{\mathcal{I}}\}\rvert = n\}$

Table 2.2 Syntax and semantics of DL role constructors: R and S are (complex) roles.

Name	Syntax	Semantics
Role negation	$\neg R$	$(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus R^{\mathcal{I}}$
Inverse	R^-	$\{(d_2,d_1) \mid (d_1,d_2) \in R^{\mathcal{I}}\}$
Composition	$R \circ S$	$\{(d_1,d_3) \mid (d_1,d_2) \in R^{\mathcal{I}}$ and $(d_2,d_3) \in S^{\mathcal{I}}\}$

Table 2.3 Syntax and semantics of TBox axioms: C_1, C_2 denote concepts and R and S roles.

Name	Syntax	Semantics
Concept inclusion	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
Role inclusion	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
Transitivity axiom	$(\mathsf{trans}\ R)$	$R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
Functionality axiom	$(\mathsf{funct}\ R)$	If $(d_1,d_2) \in R^{\mathcal{I}}$ and $(d_1,d_3) \in R^{\mathcal{I}}$, then $d_2 = d_3$

Table 2.4 Syntax and semantics of ABox assertions: a, b denote two individuals names, C a concept and R a role.

Name	Syntax	Semantics
Concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
Role assertion	$R(a,b)$	$(a^{\mathcal{I}},b^{\mathcal{I}}) \in R^{\mathcal{I}}$

Semantics

One key feature of DLs is the *open world assumption*. This means that they deal with incomplete information by considering all the possible worlds that would satisfy the axioms of the knowledge base rather than making default assumptions about unspecified information. It is a difference with the database setting that uses the closed world assumption, where the facts that are not present in the database are considered to be false. The semantics of DLs is therefore given through interpretations. In this thesis, we will make the *unique name assumption*, assuming that different individual names refer to different individuals.

Definition 2.1.4 (Interpretation)**.** An *interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set called the *domain* and $\cdot^{\mathcal{I}}$ is a function that maps each $a \in \mathsf{N_I}$ to some $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, in such a way that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ whenever $a \neq b$ (unique name assumption), each $A \in \mathsf{N_C}$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each $R \in \mathsf{N_R}$ to a set of pairs $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is extended to complex concepts and roles as specified in Tables 2.1 and 2.2.

Definition 2.1.5 (Model, entailment)**.** The satisfaction of a TBox axiom or an ABox assertion ξ in an interpretation $\mathcal{I}$, denoted $\mathcal{I} \models \xi$, is defined in Tables 2.3 and 2.4. An interpretation is a *model* of a TBox $\mathcal{T}$, written $\mathcal{I} \models \mathcal{T}$ if it satisfies every TBox axiom. It is a model of an ABox $\mathcal{A}$, written $\mathcal{I} \models \mathcal{A}$, if it satisfies every ABox assertion. Finally, it is a model of a KB $\mathcal{K}$, written $\mathcal{I} \models \mathcal{K}$, if it is a model of the ABox and of the TBox of $\mathcal{K}$.

A TBox is *satisfiable*, or *consistent*, if it has at least one model. A KB is *satisfiable*, or *consistent*, if it has at least one model. An ABox $\mathcal{A}$ is $\mathcal{T}$*-consistent* if the KB $\langle \mathcal{T}, \mathcal{A} \rangle$ is consistent. We often simply say that $\mathcal{A}$ is consistent when there is no ambiguity on the TBox considered.

A TBox axiom or ABox assertion ξ is *entailed* from a KB $\mathcal{K}$, written $\mathcal{K} \models \xi$, if $\mathcal{I} \models \xi$ for every $\mathcal{I}$ model of $\mathcal{K}$.

Query answering over description logic knowledge bases

Classical reasoning tasks over DL KBs include TBox reasoning, with *subsumption* (does $\mathcal{T} \models C_1 \sqsubseteq C_2$?) and *classification* (find all atomic concepts A, B such that $\mathcal{T} \models A \sqsubseteq B$), as well as *satisfiability*, or *consistency checking*, (is $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ satisfiable?) and *instance checking* (does $\langle \mathcal{T}, \mathcal{A} \rangle \models C(a)$?). In this thesis, we study the task of query answering, and focus on the most common query language for OMQA, namely *conjunctive queries*.

Definition 2.1.6 (Query)**.** A *first-order logic* (FOL) query over a KB is a first-order logic formula over the KB vocabulary and a set of variables. A query is called *Boolean* if it has no free variables. Given a query $q(\vec{x})$ with free variables $\vec{x} = (x_1, \dots, x_k)$ and a tuple of individuals $\vec{a} = (a_1, \dots, a_k)$, we use $q(\vec{a})$ to denote the Boolean query resulting from replacing each x_i by a_i.

Definition 2.1.7 (Conjunctive query)**.** A *conjunctive query* (CQ) is a FOL query of the form $q(\vec{x}) = \exists \vec{y} \psi(\vec{x}, \vec{y})$, where ψ is a conjunction of atoms of the forms $A(t)$ or $R(t, t')$, with t, t' individuals or variables from $\vec{x} \cup \vec{y}$. The free variables $\vec{x}$ are distinguished, or answer variables. A *union of conjunctive queries* (UCQ) is a FOL query of the form $q(\vec{x}) = \bigvee_{i=1}^{n} \exists \vec{y_i}\, \psi_i(\vec{x}, \vec{y_i})$

such that each $\exists \vec{y_i}\, \psi_i(\vec{x},\vec{y_i})$ is a CQ. A UCQ can be seen as a set of CQs. It is convenient to define the function atoms which gives the set of atoms occurring in a conjunction.

Definition 2.1.8 (Boolean query entailment)**.** A Boolean query q is satisfied in an interpretation $\mathcal{I}$, written $\mathcal{I} \models q$, iff q evaluates to true in $\mathcal{I}$. A Boolean query q is entailed by a KB $\mathcal{K}$, written $\mathcal{K} \models q$, iff q is satisfied in every model of $\mathcal{K}$.

The answers of a query are defined through entailment of Boolean queries.

Definition 2.1.9 (Answers in an interpretation)**.** A tuple $\vec{a}$ is an *answer* to a query $q(\vec{x})$ in an interpretation $\mathcal{I}$, written $\mathcal{I} \models q(\vec{a})$, iff $q(\vec{a})$ is satisfied in $\mathcal{I}$. The set of answers for $q(\vec{x})$ in $\mathcal{I}$ is written $\mathsf{ans}(q,\mathcal{I})$.

Definition 2.1.10 (Certain answers)**.** A tuple $\vec{a}$ is a *certain answer* to a query $q(\vec{x})$ over a KB $\mathcal{K}$, written $\mathcal{K} \models q(\vec{a})$, iff $\vec{a}$ is an answer to $q(\vec{x})$ in every model of $\mathcal{K}$. The set of certain answers for $q(\vec{x})$ over $\mathcal{K}$ is written $\mathsf{cert}(q,\mathcal{K})$.

It follows from the definition of certain answers that query answering over DL knowledge bases reduces straightforwardly to Boolean query entailment, since answering $q(\vec{x})$ can be done by deciding whether the Boolean query $q(\vec{a})$ is entailed by $\mathcal{K}$ for every tuple $\vec{a}$ of the same arity as $\vec{x}$.

For Boolean CQs (BCQs), we can give a finer definition of answers through the notion of *match*. A Boolean CQ $q = \exists \vec{y}\, \psi(\vec{y})$ is satisfied in an interpretation $\mathcal{I}$, iff there exists a function π, called a match, from $\vec{y}$ to $\Delta^{\mathcal{I}}$ such that $\mathcal{I} \models \xi$ for every assertion $\xi \in \mathsf{atoms}(\psi(\pi(\vec{y})))$. A match for a CQ $q(\vec{x}) = \exists \vec{y}\, \psi(\vec{x},\vec{y})$ in an interpretation $\mathcal{I}$ is therefore a function π from the variables in $\vec{x} \cup \vec{y}$ to objects in $\Delta^{\mathcal{I}}$ such that $\mathcal{I} \models \psi(\pi(\vec{x}),\pi(\vec{y}))$. By definition, $\vec{a} \in \mathsf{ans}(q,\mathcal{I})$ just in the case where there exists a match for $q(\vec{x})$ in $\mathcal{I}$ that maps each $x_i \in \vec{x}$ to $a_i \in \vec{a}$.

When we use the generic term *query* in the following, we mean a CQ.

Complexity measures

There are different complexity measures for the problem of query answering over knowledge bases, depending on what parameters of the problem are regarded as the input. *Data complexity* considers only the size of the ABox, denoted $|\mathcal{A}|$, whereas *KB complexity* considers the size of the whole KB ($|\mathcal{K}| = |\mathcal{A}| + |\mathcal{T}|$), and *combined complexity* considers the size of the whole problem, i.e. the size of the KB and the size of the query $|q|$, which is the number of atoms in the query. Data complexity is preferred when the size of the TBox and the size of the query are negligible compared to the size of the ABox, which is the case in the context of OMQA.

The following classes are utilized in this work (cf. Appendix A.1 for more details):

- AC^0: problems that can be solved by a uniform family of cicuits of constant depth and polynomial-size, with unbounded-fanin AND and OR gates.
- P: problems which are solvable in polynomial time in the size of the input.
- NP: problems which are solvable in non-deterministic polynomial time.

- coNP: problems whose complement is in NP.
- BH_2: problems that are the intersection of a problem in NP and a problem in coNP.
- Δ_2^p: problems which are solvable in polynomial time with an NP oracle.
- $\Delta_2^p[O(\log n)]$: problems which are solvable in polynomial time with at most logarithmically many calls to an NP oracle.
- Σ_2^p: problems which are solvable in non-deterministic polynomial time with an NP oracle.
- Π_2^p: problems whose complement is in Σ_2^p.

These classes are related as follow: $\mathsf{AC}^0 \subset$ P, P $\subseteq$ NP $\subseteq \Delta_2^p \subseteq \Sigma_2^p$ and P $\subseteq$ coNP $\subseteq \Delta_2^p \subseteq \Pi_2^p$, and it is widely believed that all these inclusions are proper.

2.1.2 Query answering over DL-Lite knowledge bases

As efficiency is a primary concern in OMQA, significant research efforts have been made to identify ontology languages with favorable computational properties to allow querying of large datasets. We focus on the DL-Lite family of description logics [Calvanese *et al.* 2007], and in particular on the DL-Lite$_\mathcal{R}$ dialect that underlies the OWL 2 QL profile, which is the OWL profile devoted to query answering. This family is especially interesting because query answering can be reduced to evaluation of standard database queries via query rewriting, thus benefiting from performances of modern database management systems.

Syntax of DL-Lite$_\mathcal{R}$

In DL-Lite$_\mathcal{R}$, TBox axioms are concept inclusions $B \sqsubseteq C$ and role inclusions of the form $R \sqsubseteq Q$ built according to the following syntax, where $A \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$:

$$B := A \mid \exists S, \quad C := B \mid \neg B, \quad S := R \mid R^-, \quad Q := S \mid \neg S$$

A TBox axiom of the form $B \sqsubseteq C$ or $R \sqsubseteq Q$ is a *positive inclusion*, and a TBox axiom of the form $B \sqsubseteq \neg C$ or $R \sqsubseteq \neg Q$ is a *negative inclusion*.

Example 2.1.11. We will illustrate the definitions and algorithms of the next subsections over the following DL-Lite$_\mathcal{R}$ KB about the university domain. The TBox expresses that assistant and full professor are two kinds of professors, who are persons, that those who teach are persons and that which are taught are courses, that every professor works for some entity and that working for something implies being member of that thing. Moreover, it states that assistant professors and full professors are disjoint, as are persons and courses. The ABox gives information about different individuals.

$$\begin{aligned}\mathcal{T} =&\{\mathsf{AProf} \sqsubseteq \mathsf{Prof}, \mathsf{FProf} \sqsubseteq \mathsf{Prof}, \mathsf{Prof} \sqsubseteq \mathsf{Person}, \exists \mathsf{Teach} \sqsubseteq \mathsf{Person}, \exists \mathsf{Teach}^- \sqsubseteq \mathsf{Course},\\ &\mathsf{Prof} \sqsubseteq \exists \mathsf{WorkFor}, \mathsf{WorkFor} \sqsubseteq \mathsf{MemberOf}, \mathsf{AProf} \sqsubseteq \neg \mathsf{FProf}, \mathsf{Person} \sqsubseteq \neg \mathsf{Course}\}\\ \mathcal{A} =&\{\mathsf{AProf}(ann), \mathsf{AProf}(bob), \mathsf{FProf}(bob), \mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b),\\ &\mathsf{WorksFor}(carl, dpt), \mathsf{Teach}(carl, carl)\}\end{aligned}$$

$\triangleleft$

There exist several other dialects of the DL-Lite family. In particular, DL-Lite$_{\mathsf{core}}$ is the core language of the family and amounts to DL-Lite$_{\mathcal{R}}$ without role inclusions; DL-Lite$_{\mathcal{F}}$ extends DL-Lite$_{\mathsf{core}}$ with functionality axioms on roles or on their inverses of the form $(\mathsf{funct}\ S)$; and DL-Lite$_{\mathcal{A}}$ extends DL-Lite$_{\mathsf{core}}$ with both role inclusions and functionality with the restriction that functional roles cannot be specialized, i.e. used positively on the right-hand side of a role inclusion.

Most of the results of this thesis can be extended to these other dialects, as we will see in Chapter 8. For this reason, we formulate the preliminary properties for "DL-Lite", which we will use to refer to any of the dialects DL-Lite$_{\mathsf{core}}$, DL-Lite$_{\mathcal{R}}$, DL-Lite$_{\mathcal{F}}$ or DL-Lite$_{\mathcal{A}}$.

Query answering through rewriting

A prominent approach for OMQA is query rewriting, where the query is reformulated independently from the ABox to take into account the TBox, then the reformulated query is evaluated over the ABox seen as a database. This technique allows query answering to be reduced to standard database query evaluation. However, it is not guarantee for every DL that it is possible to find such a rewriting.

We will denote by $\mathcal{I}_{\mathcal{A}} = (\Delta^{\mathcal{I}_{\mathcal{A}}}, \cdot^{\mathcal{I}_{\mathcal{A}}})$ the database-like interpretation of $\mathcal{A}$ defined as follows:

- $\Delta^{\mathcal{I}_{\mathcal{A}}} = \mathsf{Ind}(\mathcal{A})$
- $a^{\mathcal{I}_{\mathcal{A}}} = a$ for every $a \in \mathsf{Ind}(\mathcal{A})$
- $A^{\mathcal{I}_{\mathcal{A}}} = \{a \mid A(a) \in \mathcal{A}\}$ for every $A \in \mathsf{N_C}$
- $R^{\mathcal{I}_{\mathcal{A}}} = \{(a,b) \mid R(a,b) \in \mathcal{A}\}$ for every $R \in \mathsf{N_R}$

Definition 2.1.12 (FOL rewriting of a query)**.** A *FOL rewriting* of a query q w.r.t. a TBox $\mathcal{T}$ is a FOL query q' such that $\mathsf{cert}(q, \langle \mathcal{T}, \mathcal{A} \rangle) = \mathsf{ans}(q', \mathcal{I}_{\mathcal{A}})$ for all ABoxes $\mathcal{A}$. When q' is a UCQ, we call it a UCQ rewriting.

The DL-Lite family has been especially tailored to be FOL-rewritable, which means that for any satisfiable DL-Lite TBox $\mathcal{T}$ and query q, there exists a FOL-rewriting of q w.r.t. $\mathcal{T}$.

Proposition 2.1.13 (FOL-rewritability of DL-Lite)**.** *Let $\mathcal{T}$ be a satisfiable DL-Lite TBox and q a CQ. It is possible to compute a FOL rewriting q' of q in polynomial time w.r.t. the size of the TBox.*

Proposition 2.1.14 (UCQ rewriting)**.** *Let $\mathcal{T}$ be a satisfiable DL-Lite TBox and q a CQ. It is possible to compute a UCQ rewriting of q such that the size of distinct CQs in the rewriting is bounded by $O(|\mathcal{T}|^{|q|})$.*

Many algorithms have been proposed to compute FOL rewritings of CQ in DL-Lite [Pérez-Urbina *et al.* 2009, Rosati & Almatelli 2010, Calvanese *et al.* 2011, Chortaras *et al.* 2011, Venetis *et al.* 2012, Rodriguez-Muro *et al.* 2013, Gottlob *et al.* 2014], most of them producing UCQ rewritings. In the following, we assume that UCQRef is such an algorithm that produces a UCQ rewriting of a query w.r.t. a DL-Lite KB.

We illustrate rewriting algorithms with the pioneer algorithm for DL-Lite$_{\mathcal{R}}$, $\mathsf{PerfectRef}$ as it is presented in [Calvanese *et al.* 2007] (Algorithm 2.1). The general idea of the algorithm is to rewrite each atom of the query by applying the positive inclusions of the TBox, and to unify atoms when it is possible.

Definition 2.1.15 ([Calvanese *et al.* 2007])**.** The symbol "$\_$" represents non-distinguished non-shared variables. A positive inclusion I is applicable to an atom $A(x)$ if I has A in its right-hand side. A positive inclusion I is applicable to an atom $R(x,y)$ if (i) $x =\_$ and the right-hand side of I is $\exists R$, or (ii) $y =\_$ and the right-hand side of I is $\exists R^-$, or (iii) I is a role inclusion and its right-hand side is either R or R^-.

Let g be an atom and I be a positive inclusion that is applicable to g. The atom obtained from g by applying I, denoted by $gr(g,I)$, is defined as follows:

- if $g = A(x)$ and $I = A_1 \sqsubseteq A$, then $gr(g,I) = A_1(x)$
- if $g = A(x)$ and $I = \exists R \sqsubseteq A$, then $gr(g,I) = R(x,\_)$
- if $g = A(x)$ and $I = \exists R^- \sqsubseteq A$, then $gr(g,I) = R(\_,x)$
- if $g = R(x,\_)$ and $I = A \sqsubseteq \exists R$, then $gr(g,I) = A(x)$
- if $g = R(x,\_)$ and $I = \exists R_1 \sqsubseteq \exists R$, then $gr(g,I) = R_1(x,\_)$
- if $g = R(x,\_)$ and $I = \exists R_1^- \sqsubseteq \exists R$, then $gr(g,I) = R_1(\_,x)$
- if $g = R(\_,x)$ and $I = A \sqsubseteq \exists R^-$, then $gr(g,I) = A(x)$
- if $g = R(\_,x)$ and $I = \exists R_1 \sqsubseteq \exists R^-$, then $gr(g,I) = R_1(x,\_)$
- if $g = R(\_,x)$ and $I = \exists R_1^- \sqsubseteq \exists R^-$, then $gr(g,I) = R_1(\_,x)$
- if $g = R(x,y)$ and $I = R_1 \sqsubseteq R$ or $I = R_1^- \sqsubseteq R^-$, then $gr(g,I) = R_1(x,y)$
- if $g = R(x,y)$ and $I = R_1 \sqsubseteq R^-$ or $I = R_1^- \sqsubseteq R$, then $gr(g,I) = R_1(y,x)$

Example 2.1.16 (Example 2.1.11 cont'd)**.** Consider the following query that retrieves every person that is member of something: $q(x) = \exists y \mathsf{Person}(x) \wedge \mathsf{MemberOf}(x,y)$.

$\mathsf{PerfectRef}(q(x),\mathcal{T}) = \{$
$\exists y\mathsf{Person}(x) \wedge \mathsf{MemberOf}(x,y)$,
$\exists y\mathsf{Prof}(x) \wedge \mathsf{MemberOf}(x,y)$,
$\exists y\mathsf{AProf}(x) \wedge \mathsf{MemberOf}(x,y)$,
$\exists y\mathsf{FProf}(x) \wedge \mathsf{MemberOf}(x,y)$,
$\exists yz\mathsf{Teach}(x,z) \wedge \mathsf{MemberOf}(x,y)$,
$\exists y\mathsf{Person}(x) \wedge \mathsf{WorkFor}(x,y)$,
$\exists y\mathsf{Prof}(x) \wedge \mathsf{WorkFor}(x,y)$,
$\exists y\mathsf{AProf}(x) \wedge \mathsf{WorkFor}(x,y)$,
$\exists y\mathsf{FProf}(x) \wedge \mathsf{WorkFor}(x,y)$,
$\exists yz\mathsf{Teach}(x,z) \wedge \mathsf{WorkFor}(x,y)$,
$\mathsf{Person}(x) \wedge \mathsf{Prof}(x)$,
$\mathsf{Prof}(x)$,
$\mathsf{AProf}(x) \wedge \mathsf{Prof}(x)$,
$\mathsf{FProf}(x) \wedge \mathsf{Prof}(x)$,
$\exists z\mathsf{Teach}(x,z) \wedge \mathsf{Prof}(x)$,
$\mathsf{Person}(x) \wedge \mathsf{AProf}(x)$,
$\mathsf{Prof}(x) \wedge \mathsf{AProf}(x)$,
$\mathsf{AProf}(x)$,
$\mathsf{FProf}(x) \wedge \mathsf{AProf}(x)$,
$\exists z\mathsf{Teach}(x,z) \wedge \mathsf{AProf}(x)$,
$\mathsf{Person}(x) \wedge \mathsf{FProf}(x)$,
$\mathsf{Prof}(x) \wedge \mathsf{FProf}(x)$,
$\mathsf{AProf}(x) \wedge \mathsf{FProf}(x)$,
$\mathsf{FProf}(x)$,
$\exists z\mathsf{Teach}(x,z) \wedge \mathsf{FProf}(x)\}$

Algorithm 2.1 PerfectRef[Calvanese *et al.* 2007]

Input: a conjunctive query q, a TBox $\mathcal{T}$
Output: a union of conjunctive queries PR

$PR \leftarrow \{q\}$, $PR' \leftarrow \emptyset$
while $PR' \neq PR$ **do**
  $PR' \leftarrow PR$
  for all $q \in PR'$ **do**
    for all $g \in q$ **do**
      for all $I \in \mathcal{T}$ applicable to g **do**
        $PR \leftarrow PR \cup \{q[g \mapsto gr(g,I)]\}$
      end for
    end for
    for all $g_1, g_2 \in q$ **do**
      if g_1 and g_2 unify **then**
        $PR \leftarrow PR \cup \{\mathsf{reduce}(q, g_1, g_2)\}$
      end if
    end for
  end for
end while
Output PR

where reduce applies to q the most general unifier between g_1 and g_2 and replaces each unbound variable with _

$\mathsf{PerfectRef}(q(x), \mathcal{T})$ contains a lot of redundant queries. Only minimal queries, that are not contained in any other, are actually taken into account:

$\{\exists y \mathsf{Person}(x) \wedge \mathsf{MemberOf}(x,y), \quad \exists y \mathsf{Person}(x) \wedge \mathsf{WorkFor}(x,y), \quad \mathsf{Prof}(x), \quad \mathsf{FProf}(x),$
$\exists yz \mathsf{Teach}(x,z) \wedge \mathsf{MemberOf}(x,y), \quad \exists yz \mathsf{Teach}(x,z) \wedge \mathsf{WorkFor}(x,y), \quad \mathsf{AProf}(x)\}$

We detail the steps needed to produce the rewriting $\mathsf{Prof}(x)$:

1. $\mathsf{Person}(x) \wedge \mathsf{MemberOf}(x,\_)$
2. $\mathsf{Prof}(x) \wedge \mathsf{MemberOf}(x,\_)$ apply $I = \mathsf{Prof} \sqsubseteq \mathsf{Person}$
3. $\mathsf{Prof}(x) \wedge \mathsf{WorkFor}(x,\_)$ apply $I = \mathsf{WorkFor} \sqsubseteq \mathsf{MemberOf}$
4. $\mathsf{Prof}(x) \wedge \mathsf{Prof}(x)$ apply $I = \mathsf{Prof} \sqsubseteq \exists \mathsf{WorkFor}$
5. $\mathsf{Prof}(x)$ unification step

$\lhd$

Answering a CQ q over a consistent DL-Lite KB $\langle \mathcal{T}, \mathcal{A} \rangle$ amounts to searching for matches for the CQs in the UCQ rewriting of q in $\mathcal{I}_{\mathcal{A}}$. An *image* of a CQ $q(\vec{x}) = \exists \vec{y}\, \psi(\vec{x}, \vec{y})$ in $\mathcal{A}$ is a set of assertions $\mathcal{B} \subseteq \mathcal{A}$ such that there is a match $\pi : \vec{x} \cup \vec{y} \mapsto \mathsf{Ind}(\mathcal{B})$ for q in $\mathcal{I}_{\mathcal{B}}$ such that $\mathsf{atoms}(\psi(\pi(\vec{x}), \pi(\vec{y})))$ is equal to $\mathcal{B}$.

Since the first-order rewriting of the query is obtained in polynomial time in the size of the TBox, and evaluating first-order formulas over a finite interpretation is in AC^0 in the size of the interpretation, the next theorem follows:

Theorem 2.1.17 ([Calvanese *et al.* 2007, Artale *et al.* 2009])**.** *Answering UCQs in DL-Lite is in* AC^0 *in the size of the ABox (data complexity), and in* P *in the size of the TBox (so in* P *w.r.t. KB complexity).*

Since conjunctive query entailment in databases is known to be NP-complete w.r.t. combined complexity, query answering in DL-Lite is NP-hard. Membership in NP is shown in [Calvanese *et al.* 2007] by considering the non-deterministic version of PerfectRef, which non-deterministically returns only one of the conjunctive queries belonging to the reformulation of the input query. They show that this algorithm runs in NP because every query returned by PerfectRef can be generated after a polynomial number of transformations of the initial query.

Theorem 2.1.18 ([Calvanese *et al.* 2007, Artale *et al.* 2009])**.** *Answering UCQs in DL-Lite is* NP*-complete in combined complexity.*

Because of DL-Lite syntax, the rewriting of an instance query consists in a disjunction of instance queries, which can be evaluated in polynomial time by testing every possible matches.

Theorem 2.1.19. *Instance checking in DL-Lite is in* P *w.r.t. combined complexity.*

ABox consistency

We will assume throughout this thesis that the TBoxes of the KBs we consider are consistent. Indeed, the TBox is usually developed by experts and subject to extensive debugging, so it is often reasonable to assume that its content is correct, while the ABox is typically large and subject to frequent modifications, which makes errors likely.

In this case, satisfiability amounts to check whether the ABox is consistent with the TBox. In DL-Lite, ABox consistency checking reduces to answering the union of conjunctive queries corresponding to each possible violation of the TBox constraints: we can build a Boolean query $q_{unsat}^{\mathcal{T}}$ that looks for a counterexample to one of the negative inclusions or functional axioms that follow from $\mathcal{T}$, and that evaluates to true over $\mathcal{I}_{\mathcal{A}}$ just in the case that $\mathcal{A}$ is $\mathcal{T}$-inconsistent.

Example 2.1.20 (Example 2.1.11 cont'd)**.** Since $\mathcal{T}$ contains the two negative inclusions $\mathsf{AProf} \sqsubseteq \neg\mathsf{FProf}$, and $\mathsf{Person} \sqsubseteq \neg\mathsf{Course}$,

$$\begin{aligned}
q_{unsat}^{\mathcal{T}} =& \mathsf{UCQRef}(\exists x \mathsf{AProf}(x) \wedge \mathsf{FProf}(x), \mathcal{T}) \vee \mathsf{UCQRef}(\exists x \mathsf{Person}(x) \wedge \mathsf{Course}(x), \mathcal{T}) \\
=& (\exists x \mathsf{AProf}(x) \wedge \mathsf{FProf}(x)) \vee (\exists x \mathsf{Person}(x) \wedge \mathsf{Course}(x)) \vee \\
& (\exists x \mathsf{Prof}(x) \wedge \mathsf{Course}(x)) \vee (\exists x \mathsf{AProf}(x) \wedge \mathsf{Course}(x)) \vee \\
& (\exists x \mathsf{FProf}(x) \wedge \mathsf{Course}(x)) \vee (\exists xy \mathsf{Teach}(x,y) \wedge \mathsf{Course}(x)) \vee \\
& (\exists xy \mathsf{Person}(x) \wedge \mathsf{Teach}(y,x)) \vee (\exists xy \mathsf{Prof}(x) \wedge \mathsf{Teach}(y,x)) \vee \\
& (\exists xy \mathsf{AProf}(x) \wedge \mathsf{Teach}(y,x)) \vee (\exists xy \mathsf{FProf}(x) \wedge \mathsf{Teach}(y,x)) \vee \\
& (\exists xyz \mathsf{Teach}(x,y) \wedge \mathsf{Teach}(z,x)).
\end{aligned}$$

The query $q_{unsat}^{\mathcal{T}}$ evaluates to true over $\mathcal{I}_{\mathcal{A}}$ because $\mathcal{A}$ contains $\mathsf{AProf}(bob)$ and $\mathsf{FProf}(bob)$, so $x \mapsto bob$ is a match for the first CQ of $q_{unsat}^{\mathcal{T}}$ in $\mathcal{I}_{\mathcal{A}}$. $\triangleleft$

Consistency checking has therefore the same complexity as query answering.

Theorem 2.1.21. *In DL-Lite, consistency checking is* AC^0 *w.r.t. data complexity, and in* P *w.r.t. KB complexity.*

Conflicts and causes

When ABoxes are inconsistent, it is important to be able to find the reasons of the inconsistency. We define the notion of *conflicts*, that are the minimal sets of assertions responsible for the inconsistency:

Definition 2.1.22 (Conflict)**.** A *conflict* of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a minimal $\mathcal{T}$-inconsistent subset of $\mathcal{A}$. The set of conflicts of $\mathcal{K}$ is denoted $\mathsf{conflicts}(\mathcal{K})$. It can be represented as a *conflict graph*, whose vertices are assertions and which is such that there is an edge between two assertions iff they form a conflict, and a self-loop on an assertion iff it is inconsistent.

An ABox is inconsistent just in the case that its set of conflicts is non-empty. The conflicts of a DL-Lite KB can be computed while checking the consistency of the ABox with $q_{unsat}^{\mathcal{T}}$, as shown in Algorithm 2.2. Indeed, the conflicts of $\langle \mathcal{T}, \mathcal{A} \rangle$ are exactly the images of the CQs in the rewriting of $q_{unsat}^{\mathcal{T}}$ in $\mathcal{A}$ that do not contain any other image of such a CQ (e.g. if the query $\exists xyz R(x,y) \wedge R(y,z)$ has two images $\{R(a,a), R(a,b)\}$ and $\{R(a,a)\}$, only $\{R(a,a)\}$ is a conflict). The query $q_{unsat}^{\mathcal{T}}$ is constructed in lines 1 to 10 of the algorithm: for each negative (concept or role) inclusion, and each functionality axiom of $\mathcal{T}$, the conjunctive query looking for a counterexample is constructed, then rewritten with UCQRef, and the rewriting is added to $q_{unsat}^{\mathcal{T}}$. Then the CQs of $q_{unsat}^{\mathcal{T}}$ are evaluated, with all their variables considered free, so that their images can be obtained by taking for each answer the atoms of the corresponding Boolean query (lines 12 to 16). Finally, the non-minimal images are removed in order to keep only the conflicts (lines 17 to 23).

Example 2.1.23 (Example 2.1.11 cont'd)**.** The KB $\mathcal{K}$ has the following conflicts:

$$\mathsf{conflicts}(\mathcal{K}) = \{\{\mathsf{AProf}(bob), \mathsf{FProf}(bob)\}, \{\mathsf{Teach}(carl, carl)\}\}$$

$\lhd$

The fact that the CQs in $q_{unsat}^{\mathcal{T}}$ are of size two and Theorem 2.1.17 yield the following result:

Proposition 2.1.24. *Computing the conflicts of a DL-Lite KB is in* P *in data and KB complexity.*

Since in DL-Lite conflicts are of size at most two, we can define the set of *conflicts of a set of assertions* as follows:

Definition 2.1.25 (Conflicts of a set of assertions)**.** Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite KB and $\mathcal{B} \subseteq \mathcal{A}$. The set of *conflicts of* $\mathcal{B}$, denoted $\mathsf{confl}(\mathcal{B}, \mathcal{K})$, is:

$$\mathsf{confl}(\mathcal{B}, \mathcal{K}) = \{\beta \mid \exists \alpha \in \mathcal{B}, \{\alpha, \beta\} \in \mathsf{conflicts}(\mathcal{K})\} \cup \{\alpha \mid \alpha \in \mathcal{B}, \{\alpha\} \in \mathsf{conflicts}(\mathcal{K})\}.$$

Since an inconsistent KB has no model, it entails every Boolean query, and every tuple of the same arity as $\vec{x}$ is an answer to the query $q(\vec{x})$. However, only some of these answers have actually some consistent reason to hold, which is captured by the notion of *causes* of a query.

Definition 2.1.26 (Cause)**.** A *cause* for a Boolean query q in a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is an inclusion minimal $\mathcal{T}$-consistent subset $\mathcal{C} \subseteq \mathcal{A}$ such that $\langle \mathcal{T}, \mathcal{C} \rangle \models q$. We use $\mathsf{causes}(q, \mathcal{K})$ to refer to the set of causes for q in $\mathcal{K}$.

The causes for a query q in a DL-Lite KB $\langle \mathcal{T}, \mathcal{A} \rangle$ correspond to the images of the CQs of the rewriting of q in $\mathcal{A}$ that are consistent and do not contain any other image of such a CQ. Algorithm 2.3 computes the causes of a Boolean query over a DL-Lite KB. It first compute the conflicts of the KB (line 1) that will be used to check consistency of the images. The query is rewritten and evaluated over $\mathcal{A}$ with all its variables free to construct the images

Algorithm 2.2 ComputeConflicts

Input: a TBox $\mathcal{T}$, an ABox $\mathcal{A}$
Output: the conflicts of $\langle \mathcal{T}, \mathcal{A} \rangle$

$q^{\mathcal{T}}_{unsat} \leftarrow \bot$
for all $B_1 \sqsubseteq \neg B_2 \in \mathcal{T}$ **do**
  $q^{\mathcal{T}}_{unsat} \leftarrow q^{\mathcal{T}}_{unsat} \vee \mathsf{UCQRef}(\mathsf{prenex}(\exists x\, (\mathsf{at}(B_1, x) \wedge \mathsf{at}(B_2, x))), \mathcal{T})$
end for
for all $S_1 \sqsubseteq \neg S_2 \in \mathcal{T}$ **do**
  $q^{\mathcal{T}}_{unsat} \leftarrow q^{\mathcal{T}}_{unsat} \vee \mathsf{UCQRef}(\mathsf{prenex}(\exists x_1 x_2\, (\mathsf{at}(S_1, x_1, x_2) \wedge \mathsf{at}(S_2, x_1, x_2))), \mathcal{T})$
end for
for all $(\mathsf{funct}\ S) \in \mathcal{T}$ **do**
  $q^{\mathcal{T}}_{unsat} \leftarrow q^{\mathcal{T}}_{unsat} \vee \mathsf{UCQRef}(\mathsf{prenex}(\exists x x_1 x_2\, (\mathsf{at}(S, x, x_1) \wedge \mathsf{at}(S, x, x_2) \wedge x_1 \neq x_2)), \mathcal{T})$
end for
$Conflicts \leftarrow \emptyset$
for all $\exists \vec{y} \psi(\vec{y}) \in q^{\mathcal{T}}_{unsat}$ **do** // compute images of $q^{\mathcal{T}}_{unsat}$
  for all $\vec{a} \in \mathsf{ans}(\psi, \mathcal{I}_{\mathcal{A}})$ **do** // all variables in $\vec{y}$ are free here
    $Conflicts \leftarrow Conflicts \cup \{\mathsf{atoms}(\psi(\vec{a}))\}$
  end for
end for
for all $\mathcal{B} \in Conflicts$ **do** // remove non-minimal images of $q^{\mathcal{T}}_{unsat}$
  for all $\mathcal{B}' \in Conflicts$ **do**
    if $\mathcal{B}' \subseteq \mathcal{B}$ **then**
      $Conflicts \leftarrow Conflicts \backslash \mathcal{B}$
    end if
  end for
end for
Output $Conflicts$

where at is defined as follows:
$\mathsf{at}(A, x) = A(x)$
$\mathsf{at}(\exists R, x) = \exists y R(x, y)$
$\mathsf{at}(\exists R^-, x) = \exists y R(y, x)$
$\mathsf{at}(R, x_1, x_2) = R(x_1, x_2)$
$\mathsf{at}(R^-, x_1, x_2) = R(x_2, x_1)$
and prenex is defined by:
$\mathsf{prenex}(\exists \vec{x}\, (\exists \vec{y}\, g(\vec{x}, \vec{y}) \wedge \exists \vec{y'}\, g(\vec{x}, \vec{y'}))) = \exists \vec{x} \exists \vec{y} \exists \vec{z} (g(\vec{x}, \vec{y}) \wedge g(\vec{x}, \vec{z}))$ with $\vec{z} \cap \vec{y} = \emptyset$

Algorithm 2.3 ComputeCauses

Input: a Boolean conjunctive query q, a TBox $\mathcal{T}$, an ABox $\mathcal{A}$
Output: the causes for q in $\langle\mathcal{T},\mathcal{A}\rangle$

$Conflicts \leftarrow \mathsf{ComputeConflicts}(\mathcal{T},\mathcal{A})$
$Q \leftarrow \mathsf{UCQRef}(q,\mathcal{T})$
$Images \leftarrow \emptyset$
for all $\exists\vec{y}\psi(\vec{y}) \in Q$ **do** // compute images of Q
    for all $\vec{a} \in \mathsf{ans}(\psi,\mathcal{I}_{\mathcal{A}})$ **do** // all variables in $\vec{y}$ are free here
        $Images \leftarrow Images \cup \{\mathsf{atoms}(\psi(\vec{a}))\}$
    end for
end for
$Causes \leftarrow Images$
for all $\mathcal{C} \in Causes$ **do** // filter causes
    if $\exists\mathcal{B} \subseteq \mathcal{C}$, $\mathcal{B} \in Conflicts$ **then** // $\mathcal{C}$ is $\mathcal{T}$-inconsistent
        $Causes \leftarrow Causes \backslash \mathcal{C}$
    end if
end for
for all $\mathcal{C} \in Causes$ **do**
    for all $\mathcal{C}' \in Causes$ **do**
        if $\mathcal{C}' \subseteq \mathcal{C}$ **then** // $\mathcal{C}$ is not minimal
            $Causes \leftarrow Causes \backslash \mathcal{C}$
        end if
    end for
end for
Output $Causes$

(lines 4 to 8). The inconsistent or non-minimal images are then discarded (lines 10 to 21). To check the consistency of images, the algorithm simply checks that they do not contain any of the precomputed conflicts.

Example 2.1.27 (Example 2.1.11 cont'd)**.** If we call $q'(x)$ the rewriting $\mathsf{UCQRef}(q(x), \mathcal{T})$, then $q'(ann)$ has one image $\{\mathsf{AProf}(ann)\}$ in $\mathcal{A}$; $q'(bob)$ has three images $\{\mathsf{AProf}(bob)\}$, $\{\mathsf{FProf}(bob)\}$, and $\{\mathsf{Teach}(bob, c_b), \mathsf{MemberOf}(bob, dpt)\}$ in $\mathcal{A}$; and $q'(carl)$ has one image $\{\mathsf{Teach}(carl, carl), \mathsf{MemberOf}(carl, dpt)\}$.

Since $\{\mathsf{Teach}(carl, carl), \mathsf{MemberOf}(carl, dpt)\}$ is inconsistent because $\mathsf{Teach}(carl, carl)$ implies that $carl$ is both a person and a course that are disjoint, only $q(ann)$ and $q(bob)$ have causes in $\mathcal{K}$:

$$\mathsf{causes}(q(ann), \mathcal{K}) = \{\{\mathsf{AProf}(ann)\}\}$$
$$\mathsf{causes}(q(bob), \mathcal{K}) = \{\{\mathsf{AProf}(bob)\}, \{\mathsf{FProf}(bob)\}, \{\mathsf{Teach}(bob, c_b), \mathsf{MemberOf}(bob, dpt)\}\}$$

$\triangleleft$

Because of DL-Lite syntax, a cause for an atom consists of only one assertion, so the size of a cause for a query q is bounded by the size of the query $|q|$. It follows that there are at most $\sum_{j=1}^{|q|} \frac{|\mathcal{A}|\ldots(|\mathcal{A}|-j+1)}{j!} = O(|\mathcal{A}|^{|q|})$ causes, since the potential causes are the subsets of $|q|$ or fewer assertions of $\mathcal{A}$. Hence the number of potential causes and thus the number of actual causes are both polynomial in the size of $\mathcal{A}$.

Proposition 2.1.28. *Computing the causes for a query over a DL-Lite KB is in* P *w.r.t. data complexity.*

2.2 Inconsistency-tolerant semantics

When the ABox is inconsistent with the TBox, it may be impossible to clean the data to make it consistent, either for lack of time because the data is too large, or for lack of information on how to resolve the conflicts. It is therefore crucial to be able to retrieve meaningful answers from inconsistent data. However, since every query is entailed by an inconsistent KB under the classical semantics presented in the previous section, this semantics is useless in the inconsistent case, as it fails to provide any relevant information. In this section, we present *inconsistency-tolerant semantics* that have been proposed in the literature to query inconsistent KBs, focusing in particular on the AR, IAR, and brave semantics that we study in this thesis.

Example 2.2.1. The following inconsistent DL-Lite$_\mathcal{R}$ knowledge bases and query will be used to illustrate the different semantics. The TBox expresses relationships between concepts for professors (Prof) of two levels of seniority (AProf, FProf), PhD holders (PhD), postdoctoral researchers ($\mathsf{Postdoc}$), persons (Person), students ($\mathsf{Student}$) and courses (Course), and roles to link instructors to their courses (Teach), and members or employees to their organizations ($\mathsf{MemberOf}$, $\mathsf{WorkFor}$). The ABoxes provide information about different individuals.

The TBox and ABoxes are designed to illustrate the properties of all the different semantics on a small example, that is why some axioms seem not very accurate from a modelling perspective.

$$
\begin{aligned}
\mathcal{T} =& \{\mathsf{AProf} \sqsubseteq \mathsf{Prof}, \mathsf{FProf} \sqsubseteq \mathsf{Prof}, \mathsf{Prof} \sqsubseteq \mathsf{PhD}, \mathsf{Postdoc} \sqsubseteq \mathsf{PhD}, \mathsf{PhD} \sqsubseteq \mathsf{Person},\\
& \exists\mathsf{Teach} \sqsubseteq \mathsf{Person}, \exists\mathsf{Teach}^- \sqsubseteq \mathsf{Course}, \mathsf{Prof} \sqsubseteq \exists\mathsf{WorkFor}, \mathsf{Student} \sqsubseteq \exists\mathsf{MemberOf},\\
& \mathsf{WorkFor} \sqsubseteq \mathsf{MemberOf}, \mathsf{AProf} \sqsubseteq \neg\mathsf{FProf}, \mathsf{Prof} \sqsubseteq \neg\mathsf{Postdoc}, \mathsf{Student} \sqsubseteq \neg\mathsf{Prof},\\
& \mathsf{Person} \sqsubseteq \neg\mathsf{Course}, \exists\mathsf{MemberOf}^- \sqsubseteq \neg\mathsf{Postdoc}\}\\
\mathcal{A}_{ann} =& \{\mathsf{AProf}(ann), \mathsf{FProf}(ann), \mathsf{Prof}(ann), \mathsf{Teach}(ann, c_a), \mathsf{Teach}(ann, ann)\}\\
\mathcal{A}_{bob} =& \{\mathsf{AProf}(bob), \mathsf{FProf}(bob), \mathsf{Postdoc}(bob), \mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b)\}\\
\mathcal{A}_{carl} =& \{\mathsf{AProf}(carl), \mathsf{Teach}(carl, c_{c1}), \mathsf{Teach}(carl, c_{c2}), \mathsf{Teach}(c_{c1}, c_{c2}), \mathsf{Teach}(c_{c2}, c_{c1})\}\\
\mathcal{A}_{dan} =& \{\mathsf{AProf}(dan), \mathsf{Teach}(dan, c_{d1}), \mathsf{Teach}(dan, c_{d2}), \mathsf{AProf}(c_{d1}), \mathsf{AProf}(c_{d2})\}\\
\mathcal{A}_{eva} =& \{\mathsf{Student}(eva), \mathsf{AProf}(eva), \mathsf{Prof}(eva), \mathsf{Teach}(eva, c_e)\}\\
\mathcal{A}_{fred} =& \{\mathsf{Postdoc}(fred), \mathsf{MemberOf}(fred, fred), \mathsf{Teach}(fred, c_f)\}\\
q(x) =& \exists yz \mathsf{PhD}(x) \wedge \mathsf{MemberOf}(x, y) \wedge \mathsf{Teach}(x, z)
\end{aligned}
$$

All these KBs are inconsistent. Their sets of conflicts are:

$$
\begin{aligned}
\mathsf{conflicts}(\langle\mathcal{T}, \mathcal{A}_{ann}\rangle) =& \{\{\mathsf{AProf}(ann), \mathsf{FProf}(ann)\}, \{\mathsf{Teach}(ann, ann)\}\}\\
\mathsf{conflicts}(\langle\mathcal{T}, \mathcal{A}_{bob}\rangle) =& \{\{\mathsf{AProf}(bob), \mathsf{FProf}(bob)\}, \{\mathsf{AProf}(bob), \mathsf{Postdoc}(bob)\},\\
& \{\mathsf{FProf}(bob), \mathsf{Postdoc}(bob)\}\}\\
\mathsf{conflicts}(\langle\mathcal{T}, \mathcal{A}_{carl}\rangle) =& \{\{\mathsf{Teach}(carl, c_{c1}), \mathsf{Teach}(c_{c1}, c_{c2})\},\\
& \{\mathsf{Teach}(carl, c_{c2}), \mathsf{Teach}(c_{c2}, c_{c1})\},\\
& \{\mathsf{Teach}(c_{c1}, c_{c2}), \mathsf{Teach}(c_{c2}, c_{c1})\}\}\\
\mathsf{conflicts}(\langle\mathcal{T}, \mathcal{A}_{dan}\rangle) =& \{\{\mathsf{Teach}(dan, c_{d1}), \mathsf{AProf}(c_{d1})\}, \{\mathsf{Teach}(dan, c_{d2}), \mathsf{AProf}(c_{d2})\}\}\\
\mathsf{conflicts}(\langle\mathcal{T}, \mathcal{A}_{eva}\rangle) =& \{\{\mathsf{Student}(eva), \mathsf{AProf}(eva)\}, \{\mathsf{Student}(eva), \mathsf{Prof}(eva)\}\}\\
\mathsf{conflicts}(\langle\mathcal{T}, \mathcal{A}_{fred}\rangle) =& \{\{\mathsf{Postdoc}(fred), \mathsf{MemberOf}(fred, fred)\}\}
\end{aligned}
$$

$\triangleleft$

2.2.1 The AR semantics

The *AR semantics* (ABox Repair semantics) [Lembo *et al.* 2010] adapts the *consistent query answering* framework used in the database arena (cf. Chapter 7 for details and references) to DL KBs. Consistent query answering amounts to considering those answers that hold in every *repair*, defined as consistent subsets of the database that are “as close as possible” to the actual database instance. While in database setting, due to the closed world assumption and the types of constraints, repairs can be obtained from the actual database by deletion or insertion of tuples and changes of attributes values, in DL setting the inconsistency can only stem from the presence of several incompatible assertions, leading to a simpler notion of repair:

Definition 2.2.2 (Repair)**.** An *ABox repair* of a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, or repair for short, is an inclusion-maximal subset of the ABox $\mathcal{A}$ which is consistent with the TBox $\mathcal{T}$. The set of all repairs of $\mathcal{K}$ is written $\mathit{Rep}(\mathcal{T}, \mathcal{A})$.

The repairs are all the possible ways of repairing the ABox while preserving as much information as possible (in the sense of set inclusion) and can be seen as the different possible worlds. If the quality of the data is relatively good, we can assume that one of the repairs corresponds to the real world, but we cannot know which one.

Example 2.2.3 (Example 2.2.1 cont'd)**.**

$$
\begin{aligned}
\mathit{Rep}(\mathcal{T}, \mathcal{A}_{ann}) =& \{\{\mathsf{AProf}(ann), \mathsf{Prof}(ann), \mathsf{Teach}(ann, c_a)\},\\
& \{\mathsf{FProf}(ann), \mathsf{Prof}(ann), \mathsf{Teach}(ann, c_a)\}\}\\
\mathit{Rep}(\mathcal{T}, \mathcal{A}_{bob}) =& \{\{\mathsf{AProf}(bob), \mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b)\},\\
& \{\mathsf{FProf}(bob), \mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b)\},\\
& \{\mathsf{Postdoc}(bob), \mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b)\}\}\\
\mathit{Rep}(\mathcal{T}, \mathcal{A}_{carl}) =& \{\{\mathsf{AProf}(carl), \mathsf{Teach}(carl, c_{c1}), \mathsf{Teach}(carl, c_{c2})\},\\
& \{\mathsf{AProf}(carl), \mathsf{Teach}(carl, c_{c1}), \mathsf{Teach}(c_{c2}, c_{c1})\},\\
& \{\mathsf{AProf}(carl), \mathsf{Teach}(carl, c_{c2}), \mathsf{Teach}(c_{c1}, c_{c2})\}\}\\
\mathit{Rep}(\mathcal{T}, \mathcal{A}_{dan}) =& \{\{\mathsf{AProf}(dan), \mathsf{Teach}(dan, c_{d1}), \mathsf{Teach}(dan, c_{d2})\},\\
& \{\mathsf{AProf}(dan), \mathsf{Teach}(dan, c_{d1}), \mathsf{AProf}(c_{d2})\},\\
& \{\mathsf{AProf}(dan), \mathsf{Teach}(dan, c_{d2}), \mathsf{AProf}(c_{d1})\},\\
& \{\mathsf{AProf}(dan), \mathsf{AProf}(c_{d1}), \mathsf{AProf}(c_{d2})\}\}\\
\mathit{Rep}(\mathcal{T}, \mathcal{A}_{eva}) =& \{\{\mathsf{Student}(eva), \mathsf{Teach}(eva, c_e)\},\\
& \{\mathsf{AProf}(eva), \mathsf{Prof}(eva), \mathsf{Teach}(eva, c_e)\}\}\\
\mathit{Rep}(\mathcal{T}, \mathcal{A}_{fred}) =& \{\{\mathsf{Postdoc}(fred), \mathsf{Teach}(fred, c_f)\},\\
& \{\mathsf{MemberOf}(fred, fred), \mathsf{Teach}(fred, c_f)\}\}
\end{aligned}
$$

Note that the assertion $\mathsf{Teach}(ann, ann)$ does not appear in any repair of $\langle \mathcal{T}, \mathcal{A}_{ann} \rangle$ because this assertion is inconsistent. $\triangleleft$

Definition 2.2.4 (AR semantics)**.** A tuple $\vec{a}$ is an answer for a query q over a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under *AR semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{AR}} q(\vec{a})$, if and only if $\langle \mathcal{T}, \mathcal{R} \rangle \models q(\vec{a})$ for every repair $\mathcal{R} \in \mathit{Rep}(\mathcal{T}, \mathcal{A})$. We call $\vec{a}$ a (positive) *AR-answer*.

Example 2.2.5 (Example 2.2.1 cont'd)**.** The two repairs of $\mathcal{A}_{ann}$ contain the cause for $q(ann)$ $\{\mathsf{Prof}(ann), \mathsf{Teach}(ann, c_a)\}$, so $\langle \mathcal{T}, \mathcal{A}_{ann} \rangle \models_{\text{AR}} q(ann)$.

Every repair of $\mathcal{A}_{bob}$ contains a cause for $q(bob)$, and corresponds to a choice between three incompatible possibilities for *bob* being a person: he is either an associate professor

$(\{\mathsf{AProf}(bob), \mathsf{Teach}(bob, c_b)\})$, or a full professor $(\{\mathsf{FProf}(bob), \mathsf{Teach}(bob, c_b)\})$, or a post-doctoral researcher $(\{\mathsf{Postdoc}(bob), \mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b)\})$. It follows that $\langle \mathcal{T}, \mathcal{A}_{bob}\rangle \models_{\text{AR}} q(bob)$.

We have also $\langle \mathcal{T}, \mathcal{A}_{carl}\rangle \models_{\text{AR}} q(carl)$ because every repair contains one or both of the causes $\{\mathsf{AProf}(carl), \mathsf{Teach}(carl, c_{c1})\}$ and $\{\mathsf{AProf}(carl), \mathsf{Teach}(carl, c_{c2})\}$ for $q(carl)$. Here the assertions $\mathsf{Teach}(c_{c1}, c_{c2})$ and $\mathsf{Teach}(c_{c2}, c_{c1})$ implies that c_{c1} or c_{c2} may be persons and not courses, disqualifying either the first or the second cause for $q(carl)$, but not both since $\mathsf{Teach}(c_{c1}, c_{c2})$ and $\mathsf{Teach}(c_{c2}, c_{c1})$ are incompatible.

Finally, $\langle \mathcal{T}, \mathcal{A}_{dan}\rangle \not\models_{\text{AR}} q(dan)$ because the last repair of $\mathcal{A}_{dan}$ does not contain any cause for $q(dan)$ (there is no reason for dan teaching anything in that repair), $\langle \mathcal{T}, \mathcal{A}_{eva}\rangle \not\models_{\text{AR}} q(eva)$ because $q(eva)$ does not hold in the first repair where there is no reason for eva being a PhD holder, and $\langle \mathcal{T}, \mathcal{A}_{fred}\rangle \not\models_{\text{AR}} q(fred)$ because $q(fred)$ does not hold in any repair of $\mathcal{A}_{fred}$. $\triangleleft$

The AR semantics is arguably the most natural inconsistency-tolerant semantics. Unfortunately, query answering over DL-Lite KBs under AR semantics is intractable, even for instance queries, because the number of repairs may be exponential in the size of the ABox. However, it is possible to show that a query does not hold under AR semantics by non-deterministically finding a repair that does not entail it.

Theorem 2.2.6 ([Lembo *et al.* 2010]). *Conjunctive query answering and instance checking under AR semantics over DL-Lite knowledge bases is co*NP*-complete w.r.t. data complexity.*

Proof. We present the proof, which inspires proofs for several of our results.

Showing that a query q is not entailed under AR semantics by a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ can be done by guessing a repair $\mathcal{R}$ of $\mathcal{K}$ such that $\langle \mathcal{T}, \mathcal{R}\rangle \not\models q$. Checking that $\mathcal{R}$ is consistent, that for every $\alpha \in \mathcal{A}\backslash\mathcal{R}$, $\mathcal{R} \cup \{\alpha\}$ is inconsistent, and that $\langle \mathcal{T}, \mathcal{R}\rangle \not\models q$ is in P w.r.t. data complexity. Thus, query answering is in coNP.

For the lower bound, a proof for hardness by reduction from SAT is presented in [Bienvenu 2012]. Let $\varphi = \{C_1, ..., C_k\}$ be a set of clauses over a set of propositional variables $X = \{x_1, ..., x_n\}$. This problem is encoded in polynomial time as an AR entailment problem as follows:

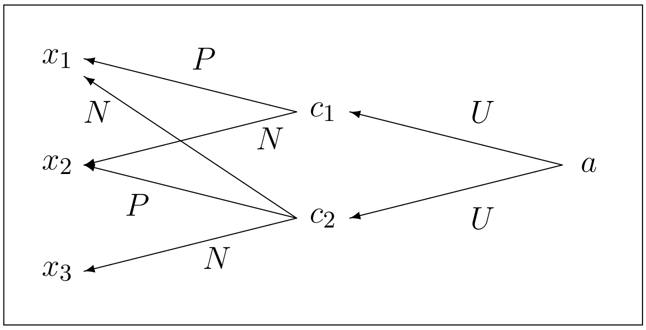
$$\begin{aligned}
\mathcal{T} =& \{\exists P^- \sqsubseteq \neg \exists N^-, \exists P \sqsubseteq \neg \exists U^-, \exists N \sqsubseteq \neg \exists U^-, \exists U \sqsubseteq A\} \\
\mathcal{A} =& \{P(c_j, x_i) \mid x_i \in C_j\} \cup \{N(c_j, x_i) \mid \neg x_i \in C_j\} \cup \{U(a, c_j) \mid 1 \leq j \leq k\} \\
q =& A(a)
\end{aligned}$$

Figure 2.1 illustrates this reduction on an example. Individuals are seen as vertices and role assertions as edges.

We show that the formula φ is satisfiable if and only if $\langle \mathcal{T}, \mathcal{A}\rangle \not\models_{\text{AR}} q$, i.e. there exists a repair that does not contain any cause for q. These causes are the $U(a, c_j)$ $(1 \leq j \leq k)$.

If φ is satisfiable, there exists a valuation ν of X such that $\nu(\varphi) = \mathsf{true}$. Thus, for every $C_j \in \varphi$, there exists $x_i \in C_j$ such that $\nu(x_i) = \mathsf{true}$ or $\neg x_i \in C_j$ such that $\nu(x_i) = \mathsf{false}$. Let $\mathcal{B} = \{P(c_j, x_i) \mid \nu(x_i) = \mathsf{true}\} \cup \{N(c_j, x_i) \mid \nu(x_i) = \mathsf{false}\}$. By construction, $\mathcal{B}$ is a consistent subset of $\mathcal{A}$ such that every individual c_j has an outgoing P- or N-edge. Therefore,

Fig. 2.1 Reduction from SAT for coNP-hardness of AR query answering. Graphical representation of the ABox constructed from an example set of clauses $\varphi = \{C_1 = x_1 \vee \neg x_2, C_2 = \neg x_1 \vee x_2 \vee \neg x_3\}$.



there exists a repair that extends $\mathcal{B}$ that does not contain any assertion of the form $U(a, c_j)$, since such an assertion would be in conflict with the edge outgoing from c_j. Since every cause for q in $\mathcal{K}$ is of that form, it follows that $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{\text{AR}} q$.

In the other direction, if $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{\text{AR}} q$, then there exists a repair $\mathcal{R}$ such that $\langle \mathcal{T}, \mathcal{R} \rangle \not\models q$. It follows that $\mathcal{R}$ does not contain any cause for q, so does not contain any assertion of the form $U(a, c_j)$. By definition of a repair, $\mathcal{R}$ is a maximal consistent subset of $\mathcal{A}$, so for each c_j, if $\mathcal{R}$ does not contain $U(a, c_j)$, it contains an assertion which is in conflict with $U(a, c_j)$, so c_j has an outgoing P- or N-edge in $\mathcal{R}$. Let ν be the valuation of X defined as follows: $\nu(x_i) = \mathsf{true}$ if there exists c_j such that $P(c_j, x_i) \in \mathcal{R}$, $\nu(x_i) = \mathsf{false}$ otherwise. For every $C_j \in \varphi$, if there exists x_i such that $P(c_j, x_i) \in \mathcal{R}$, $\nu(C_j) = \mathsf{true}$ since $x_i \in C_j$ and $\nu(x_i) = \mathsf{true}$, otherwise there exists x_i such that $N(c_j, x_i) \in \mathcal{R}$, and $\nu(C_j) = \mathsf{true}$ because $\neg x_i \in C_j$ and $\nu(x_i) = \mathsf{false}$ (since x_i cannot have an ingoing P-edge in $\mathcal{R}$, otherwise $\mathcal{R}$ would be inconsistent). $\square$

It has been shown in [Bienvenu 2012] that coNP-completeness of AR conjunctive query answering holds even for simple ontologies consisting of axioms of the forms $A_1 \sqsubseteq A_2$ and $A_1 \sqsubseteq \neg A_2$, where $A_1, A_2 \in \mathsf{N_C}$.

As in the case of classical semantics, where considering combined complexity instead of data complexity makes query answering NP-complete instead of in P, the combined complexity of CQs entailment under AR semantics is one level higher in the polynomial hierarchy than its data complexity.

Theorem 2.2.7 ([Bienvenu & Rosati 2013])**.** *Conjunctive query answering under AR semantics over DL-Lite knowledge bases is Π^p_2-complete w.r.t. combined complexity, and instance checking is co*NP*-complete w.r.t. combined complexity.*

Proof. Showing that a query q is not entailed under AR semantics can be done by guessing a repair that does not entail q. Since checking that a subset of the ABox is a repair and does not entail q is in coNP (resp. in P for instance queries) w.r.t. combined complexity, membership in Π^p_2 (resp. in coNP for instance queries) follows.

The lower bound for instance queries follows from Theorem 2.2.6. For CQs, Π^p_2-hardness is shown in [Bienvenu & Rosati 2013] by the following reduction from validity of $\text{QBF}_{2,\forall}$

formulas. Let $\varphi = \forall x_1, ..., x_n \exists y_1, ..., y_m \bigwedge_{j=1}^{k} C_j$ where $\bigwedge_{j=1}^{k} C_j$ is a 3-CNF formula over the variables $x_1, ..., x_n, y_1, ..., y_m$, where every C_j is a clause of the form $\ell_j^1 \vee \ell_j^2 \vee \ell_j^3$. The variable of literal ℓ_j^p is denoted by $v(\ell_j^p)$. For example for the clause $\neg x_1 \vee y_2 \vee \neg y_1$, $v(\neg x_1) = x_1$, $v(y_2) = y_2$, and $v(\neg y_1) = y_1$.

The problem of deciding if φ is valid is encoded in polynomial time as an AR entailment as follows:

$$
\begin{aligned}
\mathcal{T} =& \{\exists GX_i \sqsubseteq \neg \exists GX_i^- \mid 1 \leq i \leq n\} \\
\mathcal{A} =& \bigcup_{j=1}^{k} \{L_j^1(c_j^V, V(v(\ell_j^1))), L_j^2(c_j^V, V(v(\ell_j^2))), L_j^3(c_j^V, V(v(\ell_j^3))) \mid \\
& V \text{ is a valuation of } v(\ell_j^1), v(\ell_j^2), v(\ell_j^3) \text{ satisfying } C_j\} \cup \\
& \{GX_i(0,1), GX_i(1,0) \mid 1 \leq i \leq n\} \\
q =& \exists w_1, ..., w_k, x_1, ..., x_n, z_1, ..., z_n, y_1, ..., y_m \bigwedge_{j=1}^{k} \bigwedge_{h=1}^{3} L_j^h(w_j, v(\ell_j^h)) \wedge \bigwedge_{i=1}^{n} GX_i(x_i, z_i)
\end{aligned}
$$

It is shown that φ is valid if and only if $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{AR}} q$. Intuitively, $GX_i(0,1)$ means that $x_i = 0$, and the repairs of $\langle \mathcal{T}, \mathcal{A} \rangle$ correspond to the valuations of $x_1, ..., x_n$ since the conflicts of the KB are the $\{GX_i(0,1), GX_i(1,0)\}$. The query q looks for a valuation of $x_1, ..., x_n, y_1, ..., y_m$ that satisfies $\bigwedge_{j=1}^{k} C_j$: the variables $w_1, ..., w_k$ correspond to partial valuations that satisfy $C_1, ..., C_k$ respectively and will be mapped to some $c_1^{V_1}, ..., c_k^{V_k}$, and the value of $v(\ell_j^h)$ is some x_i or y_i that will be mapped to 0 or 1. $\square$

2.2.2 The IAR and brave semantics

The negative complexity results for AR semantics led [Lembo *et al.* 2010] to propose a tractable approximation of AR obtained by querying the intersection of the repairs: the *IAR semantics* (Intersection ABox Repair semantics).

Definition 2.2.8 (IAR semantics)**.** A tuple $\vec{a}$ is an answer for a query q over a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under *IAR semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{IAR}} q(\vec{a})$, if and only if $\langle \mathcal{T}, \mathcal{R}_\cap \rangle \models q(\vec{a})$ where $\mathcal{R}_\cap$ is the intersection of the repairs of $\mathcal{K}$. We call $\vec{a}$ a (positive) *IAR-answer*.

This semantics follows the "when in doubt throw it out" principle, proposed in the area of belief revision and update, and provides thus a more conservative semantics than AR. The answers holding under IAR semantics can be considered as the surest answers over the KB.

A query is entailed under IAR semantics just in the case that there exists a cause for the query in the intersection of the repairs. This condition is equivalent to the existence of a cause whose assertions do not belong to any conflict. Indeed, such a cause can be added to any consistent subset of the ABox while preserving consistency, so belongs to every repair of the ABox. In the other direction, since the conflicts are minimal inconsistent subsets of the ABox, if an assertion of a cause belongs to some conflict, the other assertions of the conflict form a consistent subset of the ABox, that can be extended to a repair that cannot contain this assertion, so does not contain this cause, which is therefore not in the intersection of the repairs.

Example 2.2.9 (Example 2.2.1 cont'd)**.** The intersections of the repairs of our examples KBs are as follows:

$$\begin{aligned}
\bigcap_{\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}_{ann})} \mathcal{R} &= \{\mathsf{Prof}(ann), \mathsf{Teach}(ann, c_a)\} \\
\bigcap_{\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}_{bob})} \mathcal{R} &= \{\mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b)\} \\
\bigcap_{\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}_{carl})} \mathcal{R} &= \{\mathsf{AProf}(carl)\} \\
\bigcap_{\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}_{dan})} \mathcal{R} &= \{\mathsf{AProf}(dan)\} \\
\bigcap_{\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}_{eva})} \mathcal{R} &= \{\mathsf{Teach}(eva, c_e)\} \\
\bigcap_{\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}_{fred})} \mathcal{R} &= \{\mathsf{Teach}(fred, c_f)\}
\end{aligned}$$

For the query q, we have only $\langle \mathcal{T}, \mathcal{A}_{ann}\rangle \models_{\text{IAR}} q(ann)$, since $\bigcap_{\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}_{ann})} \mathcal{R}$ is the only one which contains a cause for q. $\triangleleft$

FOL rewritings have been proposed for querying DL-Lite KBs under IAR semantics [Lembo *et al.* 2011, Lembo *et al.* 2015]. The general idea is to add to the classical rewriting expressions that ensure that the assertions used to derive the query are not contradicted by other assertions by enumerating the possible conflicts. For instance, using the TBox of Example 2.2.1, for the query $q(x) = \mathsf{Prof}(x)$, the standard rewriting is

$$q'(x) = \mathsf{Prof}(x) \vee \mathsf{AProf}(x) \vee \mathsf{FProf}(x),$$

and the consistent rewriting will be:

$$\begin{aligned}
q''(x) =& (\mathsf{Prof}(x) \wedge \neg\mathsf{Postdoc}(x) \wedge \neg\mathsf{Student}(x)) \vee \\
& (\mathsf{AProf}(x) \wedge \neg\mathsf{FProf}(x) \wedge \neg\mathsf{Postdoc}(x) \wedge \neg\mathsf{Student}(x)) \vee \\
& (\mathsf{FProf}(x) \wedge \neg\mathsf{AProf}(x) \wedge \neg\mathsf{Postdoc}(x) \wedge \neg\mathsf{Student}(x)).
\end{aligned}$$

It follows that querying DL-Lite KBs under IAR semantics is in AC^0 w.r.t. data complexity and in NP w.r.t. combined complexity.

Another approach consists in computing the intersection of ABox repairs in polynomial time by removing from $\mathcal{A}$ every assertion that is involved in some conflict (cf. Algorithm 2.2 for the computation of the conflicts) before querying it in a standard way. Alternatively, [Rosati *et al.* 2012] proposes to annotate the ABox, marking the assertions involved in some contradiction or self-inconsistent, and modify the standard rewriting to avoid using assertions that are not in the intersection of the repairs.

Since query answering under IAR semantics can be done through FOL rewritings, it is of the same complexity as standard query answering.

Theorem 2.2.10 ([Lembo *et al.* 2010, Lembo *et al.* 2011])**.** *Conjunctive query answering under IAR semantics over DL-Lite knowledge bases is in* AC^0 *w.r.t. data complexity,* NP-*complete w.r.t. combined complexity. Instance checking is in* P *w.r.t. combined complexity.*

While the IAR semantics is the most cautious inconsistency-tolerant semantics, the *brave semantics* [Bienvenu & Rosati 2013] at the other end of the scale considers every answer that holds in some repair.

Definition 2.2.11 (Brave semantics)**.** A tuple $\vec{a}$ is an answer for a query q over a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under *brave semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\mathsf{brave}} q(\vec{a})$, if and only if $\langle \mathcal{T}, \mathcal{R} \rangle \models q(\vec{a})$ for some repair $\mathcal{R} \in Rep(\mathcal{T}, \mathcal{A})$. We call $\vec{a}$ a (positive) *brave-answer*.

Example 2.2.12 (Example 2.2.1 cont'd)**.** Three of the repairs of $\mathcal{A}_{dan}$ contain some causes for $q(dan)$ (that are $\{\mathsf{AProf}(dan), \mathsf{Teach}(dan, c_{d1})\}$ and $\{\mathsf{AProf}(dan), \mathsf{Teach}(dan, c_{d2})\}$), so $\langle \mathcal{T}, \mathcal{A}_{dan} \rangle \models_{\mathsf{brave}} q(dan)$.

The second repair of $\mathcal{A}_{eva}$, $\{\mathsf{AProf}(eva), \mathsf{Prof}(eva), \mathsf{Teach}(eva, c_e)\}$, contains two causes for $q(eva)$, so $\langle \mathcal{T}, \mathcal{A}_{eva} \rangle \models_{\mathsf{brave}} q(eva)$.

None of the repairs of $\mathcal{A}_{fred}$ entails $q(fred)$, so $\langle \mathcal{T}, \mathcal{A}_{fred} \rangle \not\models_{\mathsf{brave}} q(fred)$. Indeed, the image of $q(fred)$ in $\mathcal{A}_{fred}$, $\{\mathsf{Postdoc}(fred), \mathsf{MemberOf}(fred, fred), \mathsf{Teach}(fred, c_f)\}$ is inconsistent so $q(fred)$ has no cause. ◁

A query is entailed under brave semantics just in the case that it is supported by some internally consistent set of facts, i.e. has at least one cause in the KB. Therefore, deciding if an answer holds under brave semantics can be done in polynomial time w.r.t. data complexity by checking if the output of Algorithm 2.3 is not empty. It is shown in [Bienvenu & Rosati 2013] that query answering under brave semantics can actually be done with FOL rewritings.

Theorem 2.2.13 ([Bienvenu & Rosati 2013])**.** *Conjunctive query answering under brave semantics over DL-Lite knowledge bases is in* AC^0 *w.r.t. data complexity,* NP-*complete w.r.t. combined complexity. Instance checking is in* P *w.r.t. combined complexity.*

The relations between the three previously introduced semantics have been shown in [Lembo *et al.* 2010, Bienvenu & Rosati 2013]: the IAR semantics is a sound approximation of the AR semantics, i.e. every query entailed under IAR is also entailed under AR semantics, whereas brave semantics can be seen as a complete approximation of AR: every query entailed under AR semantics is entailed under brave semantics.

Proposition 2.2.14. *The IAR, AR and brave semantics are related as follows, and none of the reverse implications holds:*

$$\mathcal{K} \models_{\mathit{IAR}} q(\vec{a}) \quad \Longrightarrow \quad \mathcal{K} \models_{\mathit{AR}} q(\vec{a}) \quad \Longrightarrow \quad \mathcal{K} \models_{\mathit{brave}} q(\vec{a})$$

Our examples illustrate that the reverse implications do not hold.

Example 2.2.15 (Example 2.2.1 cont'd)**.** We have seen that $\langle \mathcal{T}, \mathcal{A}_{bob} \rangle \models_{\text{AR}} q(bob)$ while $\langle \mathcal{T}, \mathcal{A}_{bob} \rangle \not\models_{\text{IAR}} q(bob)$, and that $\langle \mathcal{T}, \mathcal{A}_{dan} \rangle \models_{\mathsf{brave}} q(dan)$ while $\langle \mathcal{T}, \mathcal{A}_{dan} \rangle \not\models_{\text{AR}} q(dan)$. $\triangleleft$

Note that contrary to the AR semantics, that cannot derive contradictory statements since they have to hold in all repairs, which are consistent and can therefore not have inconsistent consequences, the answers that hold under brave semantics may be inconsistent (if they hold in different repairs).

2.2.3 Other inconsistency-tolerant semantics

The families of k-support and k-defeater semantics

Two families of inconsistency-tolerant semantics, called k-*support* and k-*defeater* semantics, have been proposed in [Bienvenu & Rosati 2013]. The k-support semantics approximate the AR semantics from below and the k-defeater semantics from above.

Definition 2.2.16 (k-support semantics)**.** A tuple $\vec{a}$ is an answer for a query q over a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under k-*support semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{k-\mathsf{supp}} q(\vec{a})$, if and only if there exists k causes $\mathcal{C}_1, ..., \mathcal{C}_k \in \mathsf{causes}(q(\vec{a}), \mathcal{K})$ such that every repair $\mathcal{R} \in Rep(\mathcal{T}, \mathcal{A})$ contains some $\mathcal{C}_i$.

Example 2.2.17 (Example 2.2.1 cont'd)**.** Since the cause $\{\mathsf{Prof}(ann), \mathsf{Teach}(ann, c_a)\}$ belongs to every repair of $\mathcal{A}_{ann}$, $\langle \mathcal{T}, \mathcal{A}_{ann} \rangle \models_{1-\mathsf{supp}} q(ann)$.

The three causes for $q(bob)$ $\{\mathsf{AProf}(bob), \mathsf{Teach}(bob, c_b)\}$, $\{\mathsf{FProf}(bob), \mathsf{Teach}(bob, c_b)\}$ and $\{\mathsf{Postdoc}(bob), \mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b)\}$ are such that each repair of $\mathcal{A}_{bob}$ contains one of them, and there is no pair of causes for $q(bob)$ that fulfills this condition, so it yields $\langle \mathcal{T}, \mathcal{A}_{bob} \rangle \models_{3-\mathsf{supp}} q(bob)$, but $\langle \mathcal{T}, \mathcal{A}_{bob} \rangle \not\models_{2-\mathsf{supp}} q(bob)$.

In the same way, $\{\mathsf{AProf}(carl), \mathsf{Teach}(carl, c_{c1})\}$ and $\{\mathsf{AProf}(carl), \mathsf{Teach}(carl, c_{c2})\}$ are sufficient and necessary to cover every repair of $\mathcal{A}_{carl}$, so $\langle \mathcal{T}, \mathcal{A}_{carl} \rangle \models_{2-\mathsf{supp}} q(carl)$, but $\langle \mathcal{T}, \mathcal{A}_{carl} \rangle \not\models_{1-\mathsf{supp}} q(carl)$. $\triangleleft$

The 1-support semantics coincides with the IAR semantics, since it requires that a cause belongs to every repair, so to their intersection. An answer holds under AR semantics just in the case where it holds under k-support semantics for some k (in particular for $k = |\mathsf{causes}(q(\vec{a}), \mathcal{K})|$).

Definition 2.2.18 (k-defeater semantics)**.** A tuple $\vec{a}$ is an answer for a query q over a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under k-*defeater semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{k-\mathsf{def}} q(\vec{a})$, if and only if there does not exist a $\mathcal{T}$-consistent subset $\mathcal{B} \subseteq \mathcal{A}$ with $|\mathcal{B}| \leq k$ such that for every cause $\mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})$, $\mathcal{B} \cup \mathcal{C}$ is $\mathcal{T}$-inconsistent.

Example 2.2.19 (Example 2.2.1 cont'd)**.** The set $\{\mathsf{AProf}(c_{d1}), \mathsf{AProf}(c_{d2})\}$ is consistent and contradicts the two causes for $q(dan)$ in $\mathcal{A}_{dan}$, and no assertion contradicts both causes, so $\langle \mathcal{T}, \mathcal{A}_{dan} \rangle \models_{1-\mathsf{def}} q(dan)$, but $\langle \mathcal{T}, \mathcal{A}_{dan} \rangle \not\models_{2-\mathsf{def}} q(dan)$.

The set $\{\mathsf{Student}(eva)\}$ is consistent and contradicts the two causes for $q(eva)$, so $\langle \mathcal{T}, \mathcal{A}_{eva} \rangle \models_{0-\mathsf{def}} q(eva)$, but $\langle \mathcal{T}, \mathcal{A}_{eva} \rangle \not\models_{1-\mathsf{def}} q(eva)$.

Finally, since there is no cause for $q(fred)$, $\langle \mathcal{T}, \mathcal{A}_{fred} \rangle \not\models_{0-\mathsf{def}} q(fred)$. $\triangleleft$

The 0-defeater semantics coincides with the brave semantics, since it only requires that the query has some cause. An answer holds under AR semantics if and only if it holds under k-defeater for every k: indeed, an answer does not hold under k-defeater for some k just in the case where there exists a $\mathcal{T}$-consistent subset $\mathcal{B} \subseteq \mathcal{A}$ such that for every cause $\mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})$, $\mathcal{B} \cup \mathcal{C}$ is $\mathcal{T}$-inconsistent, and such a subset $\mathcal{B}$ can be extended to a repair that does not contain any cause for $q(\vec{a})$.

Both semantics are monotone: for every $k \geq 0$, if $\mathcal{K} \models_{k-\mathsf{supp}} q$, then $\mathcal{K} \models_{k+1-\mathsf{supp}} q$, and if $\mathcal{K} \models_{k+1-\mathsf{def}} q$, then $\mathcal{K} \models_{k-\mathsf{def}} q$.

In DL-Lite, query answering under k-support and k-defeater semantics can be done via FOL-rewriting.

Theorem 2.2.20 ([Bienvenu & Rosati 2013]). *Conjunctive query answering under k-support and k-defeater semantics over DL-Lite knowledge bases is in* AC^0 *w.r.t. data complexity,* NP*-complete w.r.t. combined complexity. Instance checking is in* P *w.r.t. combined complexity.*

The ICR semantics

Another sound approximation of AR is the *ICR semantics* (Intersection of Closed Repairs semantics) of [Bienvenu 2012]. It achieves to be a finer approximation of AR than IAR by closing repairs with respect to the TBox before intersecting them.

Definition 2.2.21 (ICR semantics). Given a KB $\langle \mathcal{T}, \mathcal{A} \rangle$, the *logical closure* of a set of assertions $\mathcal{B}$ consists in all the assertions that are entailed by $\langle \mathcal{T}, \mathcal{B} \rangle$.

A tuple $\vec{a}$ is an answer for a query q over a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under *ICR semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{ICR}} q(\vec{a})$, if and only if $\langle \mathcal{T}, \mathcal{R}_{\cap}^{cl} \rangle \models q(\vec{a})$ where $\mathcal{R}_{\cap}^{cl}$ is the intersection of the logical closures of the repairs of $\mathcal{K}$.

The relation of ICR to AR and IAR has been shown in [Bienvenu 2012]. The intersection of the repairs clearly belongs to the intersection of their logical closures, so every IAR-answer is also an ICR-answer. Regarding AR, if an answer holds under ICR semantics, it has a cause in the intersection of the logical closures of the repairs. Every assertion of such a cause has a cause in each repair, and since the repairs are consistent, in each repair, the union of these causes is consistent. It follows that every repair contains a cause for the answer, which therefore holds under AR semantics. Actually, ICR semantics amounts to standard query answering over the assertions that hold under AR semantics.

Proposition 2.2.22. *The ICR semantics relates with IAR and AR as follows, the reverse implications do not hold:*

$$\mathcal{K} \models_{IAR} q(\vec{a}) \quad \Longrightarrow \quad \mathcal{K} \models_{ICR} q(\vec{a}) \quad \Longrightarrow \quad \mathcal{K} \models_{AR} q(\vec{a})$$

The following example illustrates that an answer can hold under the ICR semantics but not under IAR, or under AR but not under ICR.

Example 2.2.23 (Example 2.2.1 cont'd)**.** The logical closures of the repairs of $\langle \mathcal{T}, \mathcal{A}_{bob} \rangle$ are as follows:

$\{\{\mathsf{AProf}(bob), \mathsf{Prof}(bob), \mathsf{PhD}(bob), \mathsf{Person}(bob), \mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b),$
$\mathsf{Course}(c_b)\},$
$\{\mathsf{FProf}(bob), \mathsf{Prof}(bob), \mathsf{PhD}(bob), \mathsf{Person}(bob), \mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b),$
$\mathsf{Course}(c_b)\},$
$\{\mathsf{Postdoc}(bob), \mathsf{PhD}(bob), \mathsf{Person}(bob), \mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b),$
$\mathsf{Course}(c_b)\}\}$

Their intersection contains $\{\mathsf{PhD}(bob), \mathsf{MemberOf}(bob, dpt), \mathsf{Teach}(bob, c_b)\}$ that is a cause for $q(bob)$, so $\langle \mathcal{T}, \mathcal{A}_{bob} \rangle \models_{\text{ICR}} q(bob)$. However $\langle \mathcal{T}, \mathcal{A}_{bob} \rangle \not\models_{\text{IAR}} q(bob)$, since none of the causes for $\mathsf{PhD}(bob)$ is in the intersection of the repairs.

We have seen that $\langle \mathcal{T}, \mathcal{A}_{carl} \rangle \models_{\text{AR}} q(carl)$, but the intersection of the logical closures of the repairs ($\{\mathsf{AProf}(carl), \mathsf{Prof}(carl), \mathsf{PhD}(carl), \mathsf{Person}(carl)\}$) does not contain any cause for $q(carl)$, so $\langle \mathcal{T}, \mathcal{A}_{carl} \rangle \not\models_{\text{ICR}} q(carl)$. $\lhd$

Query answering under ICR semantics can be done by query rewriting for simple ontologies (consisting of axioms of the forms $A_1 \sqsubseteq A_2$ and $A_1 \sqsubseteq \neg A_2$, where $A_1, A_2 \in \mathsf{N_C}$) and even for DL-Lite without inverse roles, but it is still intractable for DL-Lite with inverse roles. Indeed, we can use the same reduction as for the proof of Theorem 2.2.6 for AR coNP-hardness.

Theorem 2.2.24 ([Bienvenu 2012, Lukasiewicz *et al.* 2013])**.** *Conjunctive query answering and instance checking under ICR semantics over DL-Lite knowledge bases is co*NP*-complete w.r.t. data complexity. Instance checking is co*NP*-complete w.r.t. combined complexity.*

However, we next show that regarding combined complexity and CQs, ICR is easier than AR semantics.

Theorem 2.2.25. *Conjunctive query answering under ICR semantics over DL-Lite knowledge bases is $\Delta_2^p[O(log\ n)]$-complete w.r.t. combined complexity.*

Proof. To decide if a query is entailed under ICR semantics, we first use a coNP-oracle to decide, for each assertion built using the KB signature and individuals, whether the assertion is entailed under AR semantics. Then we use a NP-oracle to decide if the query holds w.r.t. the ABox consisting of all assertions that are AR-entailed. Since the oracle calls can be structured as a tree, this procedure gives membership in $\Delta_2^p[O(\text{log n})]$ (cf. Appendix A.1, [Gottlob 1995]).

For the lower bound, the proof is by reduction from the Parity(3SAT) problem, cf. [Wagner 1987, Eiter & Gottlob 1997]. A Parity(3SAT) instance is given by a sequence $\varphi_1, \ldots, \varphi_m$ of propositional formulas in 3CNF, and the problem is to decide whether the number of satisfiable formulas is odd. It is known that it can be assumed w.l.o.g. that the formulas are such that φ_{i+1} is unsatisfiable whenever φ_i is unsatisfiable. Consequently, the problem reduces to deciding existence of an odd integer p such that φ_p is satisfiable and φ_{p+1} is unsatisfiable.

Consider a Parity(3SAT) instance given by $\varphi_1, \dots, \varphi_m$. For each i, $\varphi_i = \bigwedge_{j=1}^{k(i)} C_{i,j}$, where every $C_{i,j}$ is a clause of the form $\ell^1_{i,j} \vee \ell^2_{i,j} \vee \ell^3_{i,j}$ over variables $X_i = \{x_{i,1}, ..., x_{i,n(i)}\}$. The variable of literal $\ell^p_{i,j}$ is denoted by $v(\ell^p_{i,j})$. Let $k = \mathsf{max}_{1\leq i\leq m}(k(i))$ and $n = \mathsf{max}_{1\leq i\leq m}(n(i)) + 1$. We can assume that $k(i) = k$ and $n(i) = n$ for every i by adding $k - k(i)$ clauses of the form $x_{i,n} \vee x_{i,n} \vee x_{i,n}$ to each φ_i without changing φ_i (un)satisfiability. We define an ICR entailment problem as follows:

$$
\begin{aligned}
\mathcal{T} =& \{\exists P^- \sqsubseteq \neg\exists N^-, \exists P \sqsubseteq \neg\exists U^-, \exists N \sqsubseteq \neg\exists U^-, \exists U \sqsubseteq A\}\\
\mathcal{A}_o(i) =& \{\bigcup_{j=1}^{k} \{R_i(\varphi_i, c^V_{i,j}), L^1_j(c^V_{i,j}, V(v(\ell^1_{i,j}))), L^2_j(c^V_{i,j}, V(v(\ell^2_{i,j}))), L^3_j(c^V_{i,j}, V(v(\ell^3_{i,j}))) \mid\\
& \quad V \text{ is a valuation of } v(\ell^1_{i,j}), v(\ell^2_{i,j}), v(\ell^3_{i,j}) \text{ satisfying } C_{i,j}\}\\
\mathcal{A}_e(i) =& \{P(c_{i,j}, x_{i,l}) \mid x_{i,l} \in C_{i,j}\} \cup \{N(c_{i,j}, x_{i,l}) \mid \neg x_{i,l} \in C_{i,j}\} \cup \{U(\varphi_i, c_{i,j}) \mid 1 \leq j \leq k\}\\
\mathcal{A}'_e(i) =& \{E(a_i, \varphi_i)\} \cup \{R_f(a_i, b_i) \mid 1 \leq f \leq m, f \text{ odd }, f \neq i-1\} \cup\\
& \{L^1_j(b_i, c_i), L^2_j(b_i, c_i), L^3_j(b_i, c_i) \mid 1 \leq j \leq k\}\\
\mathcal{A} =& \bigcup_{1\leq i\leq m, i \text{ even}} (\mathcal{A}_e(i) \cup \mathcal{A}'_e(i)) \cup \{E(\varphi_i, \varphi_{i+1}) \mid 1 \leq i \leq m-1, i \text{ odd}\}\\
& \cup \bigcup_{1\leq i\leq m, i \text{ odd}} \mathcal{A}_o(i) \cup \{A(\varphi_{m+1}), E(\varphi_m, \varphi_{m+1}) \mid \text{if } m \text{ is odd}\} \cup \mathcal{A}'_e(m+1)\\
q =& A(x) \wedge \bigwedge_{i=1,\text{odd}}^{m} (E(y_i, x) \wedge \bigwedge_{j=1}^{k} (R_i(y_i, w_{i,j}) \wedge \bigwedge_{h=1}^{3} L^h_j(w_{i,j}, v(\ell^h_{i,j})))),
\end{aligned}
$$

every variable of q being existentially quantified.

Figure 2.2 illustrates the ABox of this reduction. We show that $\langle\mathcal{T}, \mathcal{A}\rangle \models_{\text{ICR}} q$ iff there is some odd p such that φ_p is satisfiable and φ_{p+1} is unsatisfiable (or if φ_m is satisfiable and m is odd).

For the first direction, suppose that $\langle\mathcal{T}, \mathcal{A}\rangle \models_{\text{ICR}} q$. This means that there exists a match π for q in the intersection of the logical closures of the repairs. By construction, $\pi(x) = \varphi_{p+1}$ for some odd p, since only φ_i with even i can belong to A.

If $p \neq m$, for the same reasons as in the proof of Theorem 2.2.6, $A(\varphi_{p+1})$ is entailed under ICR semantics iff φ_{p+1} is unsatisfiable, so φ_{p+1} is unsatisfiable. Since φ_{p+1} is linked by incoming E-edges to φ_p and a_{p+1}, and φ_p has outgoing R_p-edges while a_{p+1} has outgoing R_i-edges for every odd $i \neq p$, then for every odd $i \neq p$, $\pi(y_i) = a_{p+1}$, $\pi(w_{i,j}) = b_{p+1}$ and $\pi(x_{i,l}) = c_{p+1}$ for every $1 \leq j \leq k$ and $1 \leq l \leq n$, and $\pi(y_p) = \varphi_p$. Then for every $1 \leq j \leq k$, we have $R_p(\varphi_p, \pi(w_{p,j})) \in \mathcal{A}$, so there is a valuation V_j of $v(\ell^1_{p,j}), v(\ell^2_{p,j}), v(\ell^3_{p,j})$ satisfying $C_{p,j}$ such that $\pi(w_{p,j}) = c^{V_j}_{p,j}$. Note that for every $x_{p,l}$, either $\pi(x_{p,l}) = 0$ or $\pi(x_{p,l}) = 1$, so for every $1 \leq j, j' \leq k$, V_j and $V_{j'}$ must agree on the value of $x_{p,l}$. It follows that the V_j together define a valuation of the $x_{p,l}$ which satisfies φ_p. Thus there is an odd p such that φ_p is satisfiable and φ_{p+1} is unsatisfiable.

In the case where $p = m$, $A(\varphi_{m+1})$ has no conflicts so belongs to every repair. Then as in the case $p \neq m$, φ_m is satisfiable. Thus there is an odd number of satisfiable formulas.

In the other direction, suppose that there is an odd $p \neq m$ such that φ_p is satisfiable and φ_{p+1} is unsatisfiable. Let ν be a valuation of $x_{p,1},...,x_{p,n}$ that satisfies φ_p. Let π be the function defined by: $\pi(y_p) = \varphi_p$, $\pi(x) = \varphi_{p+1}$, for every $1 \leq j \leq k$, $\pi(w_{p,j}) = c_{p,j}^{V_j}$, where V_j is the restriction of ν to the variables in clause $c_{p,j}$, for every $1 \leq l \leq n$, $\pi(x_{p,l}) = \nu(x_{p,l})$, and for every odd $i \neq p$: $\pi(y_i) = a_{p+1}$, for every $1 \leq j \leq k$, $\pi(w_{i,j}) = b_{p+1}$, for every $1 \leq l \leq n$, $\pi(x_{i,l}) = c_{p+1}$. Then π is a match for q in the intersection of the logical closures of the repairs. Indeed, the only assertion of the query obtained by replacing the variables by their images by π which has some conflicts so is not already in the intersection of the repairs is $A(\varphi_{p+1})$, which is ICR because φ_{p+1} is unsatisfiable. It follows that q is entailed under ICR semantics.

Finally, if φ_m is satisfiable and m is odd, we can find a match for q using a valuation that satisfies φ_m as in the case $p \neq m$. In this case, since $A(\varphi_{m+1})$ has no conflicts, it is also in the intersection of the repairs and q holds under ICR semantics (actually it holds under IAR semantics). □

The CAR and ICAR semantics

The idea of logically closing part of the data w.r.t. the TBox has also been considered in [Lembo *et al.* 2010], with the *CAR* (Closed ABox Repair) and *ICAR semantics* (Intersection Closed ABox Repair semantics). Contrary to the ICR semantics which closes the repairs, the CAR and ICAR semantics close directly the ABox by adding every assertion that is entailed by a consistent subset of the ABox, i.e. which has some cause. CAR and ICAR are then defined similarly to AR and IAR.

Definition 2.2.26 (CAR and ICAR semantics)**.** The *consistent logical consequences* of $\mathcal{K}$ is the set $clc(\mathcal{K}) = \{\alpha \mid \text{ there exists } \mathcal{B} \subseteq \mathcal{A} \text{ such that } \mathcal{B} \text{ is } \mathcal{T}\text{-consistent and } \langle \mathcal{T}, \mathcal{B} \rangle \models \alpha\}$.

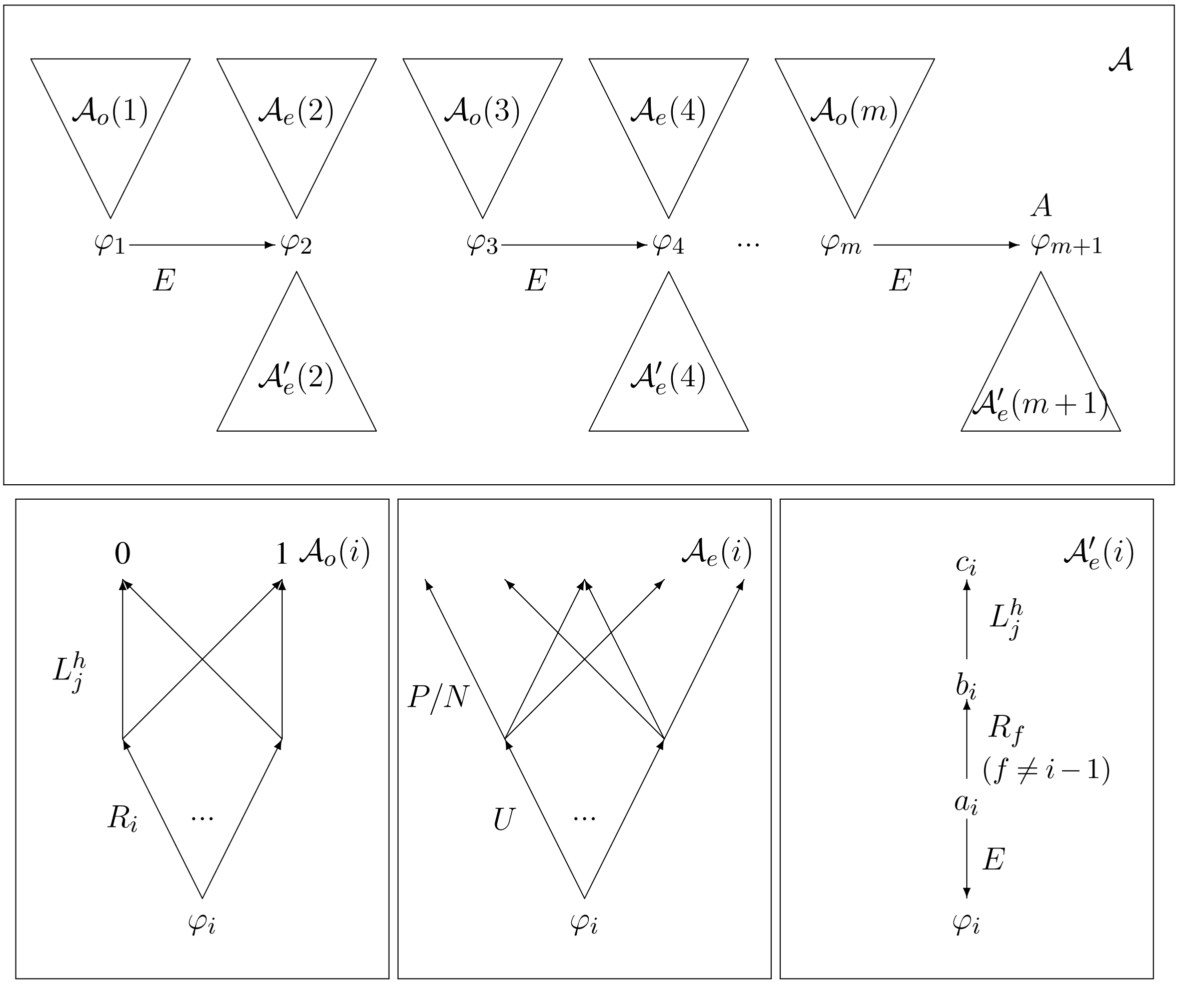
A *closed ABox repair* $\mathcal{R}^{clc}$ is a consistent subset of $clc(\mathcal{K})$ such that there exists no consistent subset $\mathcal{R}^{clc'}$ of $clc(\mathcal{K})$ such that $\mathcal{R}^{clc} \cap \mathcal{A} \subset \mathcal{R}^{clc'} \cap \mathcal{A}$ or $\mathcal{R}^{clc} \cap \mathcal{A} = \mathcal{R}^{clc'} \cap \mathcal{A}$ and $\mathcal{R}^{clc} \subset \mathcal{R}^{clc'}$.

A tuple $\vec{a}$ is an answer for a query q over a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under *CAR semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{CAR}} q(\vec{a})$, if and only if $\langle \mathcal{T}, \mathcal{R}^{clc} \rangle \models q(\vec{a})$ for every closed ABox repair $\mathcal{R}^{clc}$ of $\mathcal{K}$.

A tuple $\vec{a}$ is an answer for a query q over a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under *ICAR semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{ICAR}} q(\vec{a})$, if and only if $\langle \mathcal{T}, \mathcal{R}^{clc}_{\cap} \rangle \models q(\vec{a})$ where $\mathcal{R}^{clc}_{\cap}$ is the intersection of the closed ABox repairs of $\mathcal{K}$.

Remark 2.2.27. It is equivalent to define the ICAR semantics with the intersection of the repairs of $clc(\mathcal{K})$: indeed, every closed ABox repair is a repair of $clc(\mathcal{K})$, so the intersection of the repairs of $clc(\mathcal{K})$ is included in $\mathcal{R}^{clc}_{\cap}$, and in the other direction, if $\alpha \in \mathcal{R}^{clc}_{\cap}$, then α is in the intersection of the repairs of $clc(\mathcal{K})$, otherwise there would be $\beta \in clc(\mathcal{K})$ that conflicts α, and α would also be in a conflict with a cause $\gamma \in \mathcal{A}$ of β, so some closed ABox repair would contain γ, that would yield $\alpha \notin \mathcal{R}^{clc}_{\cap}$.

Fig. 2.2 Reduction from Parity(3SAT) for $\Delta_2^p[O(\log \mathrm{n})]$-hardness of ICR query answering. Graphical representation for the case where m is odd.



This is not true for the CAR semantics since some repairs of $clc(\mathcal{K})$ may not be closed ABox repairs. For instance, let $\mathcal{T} = \{R \sqsubseteq S, \exists R \sqsubseteq \neg\exists R^-, \exists S \sqsubseteq \neg\exists R^-, \exists R \sqsubseteq \neg\exists S^-\}$, $\mathcal{A} = \{R(a,b), R(b,c)\}$ and $q = \exists xy R(x,y)$. The set $\{S(a,b), S(b,c)\}$ is a repair of $clc(\langle\mathcal{T},\mathcal{A}\rangle)$ such that $\langle\mathcal{T}, \{S(a,b), S(b,c)\}\rangle \not\models q$, but $\langle\mathcal{T},\mathcal{A}\rangle \models_{\text{CAR}} q$ because it is not a closed ABox repair, since contrary to the other repairs of $clc(\langle\mathcal{T},\mathcal{A}\rangle)$ it does not contain any assertion of $\mathcal{A}$.

The CAR and ICAR semantics are not sound approximations of the AR semantics: some queries may hold under CAR or ICAR and not under AR semantics. [Lembo *et al.* 2010] motivates these semantics arguing that if $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ and $\mathcal{K}' = \langle\mathcal{T},\mathcal{A}'\rangle$ differ only because $\mathcal{A}'$ contains assertions that are logical consequences of consistent subsets of $\mathcal{A}$, they should yield the same query answers, which is not true under AR semantics. These semantics enforce the consistency of the set of answers retrieved, but query answers may have no cause in the initial knowledge base.

The relations between ICAR, CAR and AR have been shown in [Lembo *et al.* 2010]. ICAR and CAR relate similarly to IAR and AR, and every AR-answer holds under CAR semantics because by definition, every closed ABox repair contains a repair.

Proposition 2.2.28. *The CAR and ICAR semantics relate with ICR and AR as follows, the reverse implications do not hold:*

$$\mathcal{K} \models_{ICR} q(\vec{a}) \implies \mathcal{K} \models_{ICAR} q(\vec{a}) \implies \mathcal{K} \models_{CAR} q(\vec{a}) \text{ and } \mathcal{K} \models_{AR} q(\vec{a}) \implies \mathcal{K} \models_{CAR} q(\vec{a})$$

Proof of $\mathcal{K} \models_{ICR} q(\vec{a}) \implies \mathcal{K} \models_{ICAR} q(\vec{a})$. The relation between ICR and ICAR semantics follows from the fact that the intersection of the logical closure of the repairs of $\mathcal{K}$ is included in the intersection of the repairs of the consistent logical consequences of $\mathcal{K}$: $\mathcal{R}^{cl}_{\cap} \subseteq \mathcal{R}^{clc}_{\cap}$. Indeed, suppose for a contradiction that $\alpha \in \mathcal{R}^{cl}_{\cap}$ and $\alpha \notin \mathcal{R}^{clc}_{\cap}$. Since α is in the logical closure of every repair, α is a consistent consequence of $\mathcal{K}$, so if $\alpha \notin \mathcal{R}^{clc}_{\cap}$, that means that there exists $\beta \in clc(\mathcal{K})$ such that $\{\alpha,\beta\}$ is inconsistent. Since β is entailed by a consistent subset of $\mathcal{A}$, which can be extended to a repair, β is in the logical closure of a repair $\mathcal{R}^{cl}_{\beta}$. The logical closure of a repair is consistent, so $\alpha \notin \mathcal{R}^{cl}_{\beta}$, so $\alpha \notin \mathcal{R}^{cl}_{\cap}$. □

These relations can be observed in our example KBs.

Example 2.2.29 (Example 2.2.1 cont'd)**.** For the KB $\langle\mathcal{T},\mathcal{A}_{eva}\rangle$, we have

$$\begin{aligned} clc(\langle\mathcal{T},\mathcal{A}_{eva}\rangle) = \{&\mathsf{Student}(eva), \mathsf{AProf}(eva), \mathsf{Prof}(eva), \mathsf{PhD}(eva), \mathsf{Person}(eva),\\ &\mathsf{Teach}(eva, c_e), \mathsf{Course}(c_e)\}\end{aligned}$$

So the closed ABox repairs of $\langle\mathcal{T},\mathcal{A}_{eva}\rangle$ are:

$$\begin{aligned}&\{\{\mathsf{Student}(eva), \mathsf{PhD}(eva), \mathsf{Person}(eva), \mathsf{Teach}(eva, c_e), \mathsf{Course}(c_e)\},\\ &\{\mathsf{AProf}(eva), \mathsf{Prof}(eva), \mathsf{PhD}(eva), \mathsf{Person}(eva), \mathsf{Teach}(eva, c_e), \mathsf{Course}(c_e)\}\}\end{aligned}$$

Since $\mathsf{Student} \sqsubseteq \exists\mathsf{MemberOf}$ and $\mathsf{Prof} \sqsubseteq \exists\mathsf{WorkFor}$, $\langle\mathcal{T},\mathcal{A}_{eva}\rangle \models_{\text{CAR}} q(eva)$. The intersection of the closed ABox repairs does not contain any cause for $q(eva)$, so $\langle\mathcal{T},\mathcal{A}_{eva}\rangle \not\models_{\text{ICAR}} q(eva)$. We have seen that $\langle\mathcal{T},\mathcal{A}_{eva}\rangle \not\models_{\text{AR}} q(eva)$.

For the KB $\langle\mathcal{T},\mathcal{A}_{fred}\rangle$, we have

$$\begin{aligned}clc(\langle\mathcal{T},\mathcal{A}_{fred}\rangle) = &\{\mathsf{Postdoc}(fred), \mathsf{PhD}(fred), \mathsf{Person}(fred),\\ &\mathsf{MemberOf}(fred, fred), \mathsf{Teach}(fred, c_f), \mathsf{Course}(c_f)\}\end{aligned}$$

So the closed ABox repairs of $\langle\mathcal{T},\mathcal{A}_{fred}\rangle$ are:

$$\{\{\mathsf{Postdoc}(fred), \mathsf{PhD}(fred), \mathsf{Person}(fred), \mathsf{Teach}(fred, c_f), \mathsf{Course}(c_f)\},$$
$$\{\mathsf{PhD}(fred), \mathsf{Person}(fred), \mathsf{MemberOf}(fred, fred), \mathsf{Teach}(fred, c_f), \mathsf{Course}(c_f)\}\}$$

It follows that $\langle\mathcal{T},\mathcal{A}_{fred}\rangle \models_{\text{ICAR}} q(fred)$. Note that $\langle\mathcal{T},\mathcal{A}_{fred}\rangle \not\models_{\mathsf{brave}} q(fred)$. ICAR semantics considers as true every consequence of the data that is not contradicted, so allows to retrieve such answers that can only be derived from inconsistent images in the original ABox as long as nothing contradicts them.

Finally, we can note that $\langle\mathcal{T},\mathcal{A}_{dan}\rangle \not\models_{\text{CAR}} q(dan)$ because the closed ABox repair that contains $\mathsf{AProf}(c_{d_1})$, $\mathsf{AProf}(c_{d_2})$ and their consequences does not contain any cause for $q(dan)$. ◁

[Lembo *et al.* 2011] describes how to query DL-Lite KBs under ICAR semantics using FOL-rewriting. Moreover, instance checking under CAR semantics reduces to checking if the assertion belongs to the intersection of the closed repairs, i.e. to ICAR semantics.

Theorem 2.2.30 ([Lembo *et al.* 2010, Lembo *et al.* 2011])**.** *Conjunctive query answering under ICAR semantics over DL-Lite knowledge bases is in* AC^0 *w.r.t. data complexity. Conjunctive query answering under CAR semantics is co*NP*-complete w.r.t. data complexity, and instance checking is in* AC^0 *w.r.t. data complexity. Conjunctive query answering under ICAR semantics is* NP*-complete w.r.t. combined complexity. Instance checking under CAR and ICAR semantics is in* P *w.r.t. combined complexity.*

Theorem 2.2.31. *Conjunctive query answering under CAR semantics is* Π_2^p*-complete w.r.t. combined complexity.*

Proof. Computing $clc(\mathcal{K})$ is in P w.r.t. combined complexity and checking that a subset of $clc(\mathcal{K})$ is a closed ABox repair that does not entail q is in Δ_2^p: check in P that it is consistent, in coNP that it is maximal, and in coNP that it preserves a maximal subset of the ABox, then check that it does not entail q in coNP. It follows that CQ answering under CAR is in Π_2^p.

The reduction given for Π_2^p-hardness of AR CQ entailment in proof of Theorem 2.2.7 can be used to show Π_2^p-hardness. Indeed, since $\mathcal{T}$ does not contain any positive inclusion, $clc(\mathcal{K}) = \mathcal{K}$ and q is entailed under CAR semantics if and only if it is entailed under AR semantics. □

The family of k-lazy semantics

A family of inconsistency-tolerant semantics, the k-lazy semantics, has been introduced in [Lukasiewicz *et al.* 2012]. They are based on the notion of k-lazy repairs that are obtained from the ABox by removing for each *cluster* of conflicts, i.e. sets of assertions that belong to some conflict that share assertions, either a minimal subset of at most k assertions that restores the consistency of the cluster, or the whole cluster if there is no such subset. Note that not every lazy repair is a repair because of the possible removal of some entire clusters.

Definition 2.2.32 (k-lazy semantics)**.** The *clusters* are the connected components of the conflict graph that contain at least one edge.

A *k-lazy repair* of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is obtained by removing from $\mathcal{A}$ for each cluster: (i) either an inclusion minimal set that restores the consistency of the cluster and is of size at most k, (ii) or the whole cluster if there does not exist such a set.

A tuple $\vec{a}$ is an answer for a query q over a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under *k-lazy semantics*, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{k-\mathsf{lazy}} q(\vec{a})$, if and only if $\langle \mathcal{T}, \mathcal{R}_k \rangle \models q(\vec{a})$ for every k-lazy repair $\mathcal{R}_k$ of $\mathcal{K}$.

Example 2.2.33 (Example 2.2.1 cont'd)**.** There is only one 0-lazy repair for $\langle \mathcal{T}, \mathcal{A}_{ann} \rangle$, obtained by removing every cluster, so every assertion that belongs to some conflict. It follows that it corresponds to the intersection of the repairs, so $\langle \mathcal{T}, \mathcal{A}_{ann} \rangle \models_{0-\mathsf{lazy}} q(ann)$.

To restore the consistency of the cluster $\{\mathsf{AProf}(bob), \mathsf{FProf}(bob), \mathsf{Postdoc}(bob)\}$, it is sufficient and necessary to remove two assertions. There are three 2-lazy repairs, that all contain some cause for $q(bob)$. It follows that $\langle \mathcal{T}, \mathcal{A}_{bob} \rangle \models_{k-\mathsf{lazy}} q(bob)$ for $k \geq 2$, but $\langle \mathcal{T}, \mathcal{A}_{bob} \rangle \not\models_{1-\mathsf{lazy}} q(bob)$ since the 1-lazy repair is obtained by removing the whole cluster, so there is no cause left in this repair.

Similarly, to restore the consistency of the cluster $\{\mathsf{Teach}(carl, c_{c1}), \mathsf{Teach}(carl, c_{c2}), \mathsf{Teach}(c_{c1}, c_{c2}), \mathsf{Teach}(c_{c2}, c_{c1})\}$ it is sufficient and necessary to remove two assertions and the obtained 2-lazy repairs contain causes for $q(carl)$, whereas there is no cause in the 1-lazy repair. It follows that $\langle \mathcal{T}, \mathcal{A}_{carl} \rangle \models_{k-\mathsf{lazy}} q(carl)$ for $k \geq 2$, and $\langle \mathcal{T}, \mathcal{A}_{carl} \rangle \not\models_{1-\mathsf{lazy}} q(carl)$.

Finally, for $\langle \mathcal{T}, \mathcal{A}_{eva} \rangle$, the only way of repairing the cluster $\{\mathsf{Student}(eva), \mathsf{AProf}(eva), \mathsf{Prof}(eva)\}$ by removing one assertion is to remove $\mathsf{Student}(eva)$, and the 1-lazy repair contains causes for $q(eva)$, so $\langle \mathcal{T}, \mathcal{A}_{eva} \rangle \models_{1-\mathsf{lazy}} q(eva)$, but there are two k-lazy repairs for $k \geq 2$: $\{\mathsf{AProf}(eva), \mathsf{Prof}(eva), \mathsf{Teach}(eva, c_e)\}$ and $\{\mathsf{Student}(eva), \mathsf{Teach}(eva, c_e)\}$, and the second one does not contain any cause for $q(eva)$, so $\langle \mathcal{T}, \mathcal{A}_{eva} \rangle \not\models_{k-\mathsf{lazy}} q(eva)$ for $k \geq 2$. $\triangleleft$

An answer holds under IAR semantics if and only if it holds under 0-lazy semantics, that implies that it also hold under every k-lazy semantics. The k-lazy semantics are not a sound approximation of AR but a compromise between IAR and brave, and it is notable that these semantics are not monotone in k as shown in the preceeding example (eva case). However, for every KB $\mathcal{K}$ and query q, there exists k such that $\mathcal{K} \models_{\mathsf{AR}} q$ iff $\mathcal{K} \models_{k-\mathsf{lazy}} q$.

Theorem 2.2.34 ([Lukasiewicz *et al.* 2012, Bienvenu 2012])**.** *Conjunctive query answering under k-lazy semantics over DL-Lite knowledge bases is co*NP*-complete w.r.t. data complexity for every $k \geq 1$. Instance checking is in* P *w.r.t. data complexity.*

Theorem 2.2.35. *Conjunctive query answering under k-lazy semantics semantics is Π_2^p-complete w.r.t. combined complexity for every $k \geq 1$.*

Proof. Checking that a subset of $\mathcal{A}$ is k-lazy repair that does not entail q is in Δ_2^p: check consistency in P, check in coNP that it is impossible to add an assertion of a cluster that has not be fully removed without loosing the consistency, and check in coNP that it is not possible to add one of the fully removed cluster except k assertions, then check that it does not entail q in coNP. It follows that CQ answering under k-lazy semantics is in Π_2^p.

The reduction given for Π_2^p-hardness of AR CQ entailment in proof of Theorem 2.2.7 can be used to show Π_2^p-hardness. Indeed, the clusters of conflicts are exactly the conflicts (the $\{GX_i(0,1), GX_i(1,0)\}$, for $1 \leq i \leq n$), so are of size two and can be repaired by removing exactly one assertion, so the repairs correspond to the k-lazy repairs for every $k \geq 1$, and q is entailed under the k-lazy semantics for $k \geq 1$ if and only if it is entailed under AR semantics. □

The general modifier-based framework for inconsistency-tolerant query answering

Recently, [Baget *et al.* 2016] introduced a general framework that captures many of the preceeding semantics and creates new ones. An inconsistency-tolerant semantics is defined as a pair composed of a modifier, that creates a set of ABoxes from the original ABox, and an inference strategy to derive queries from these ABoxes. The paper studies in particular the semantics that use complex modifiers build from the three basic modifiers *positive closure* (C) that adds to each ABox all assertions that can be derived by applying the positive inclusions of the TBox, *splitting into repairs* (R) of each ABox, and *selecting the cardinality-maximal* (M) ABoxes, and one of the four inference strategies *universal* ($\forall$) that considers true the conclusions that are entailed by every ABox, *safe* ($\cap$) that considers true the conclusions that are entailed by the intersection of the ABoxes, *majority-based* (maj) that considers true the conclusions that are entailed by the majority of the ABoxes, and *existential* ($\exists$) that considers true the conclusions that are entailed by some ABox. This framework can be extended with other modifiers or inference strategies to cover more semantics.

The AR semantics corresponds to the so-called $\langle R, \forall \rangle$, that is, using the modifier splitting into repairs and the universal inference strategy. The IAR semantics corresponds to $\langle R, \cap \rangle$, and the brave semantics to the $\langle R, \exists \rangle$. The ICR semantics corresponds to $\langle CR, \cap \rangle$, the modifier being the composition of positive closure with splitting into repairs. The ICAR, CAR and k-lazy semantics can be included in this general framework by defining other modifiers (adding consistent closure is sufficient to capture ICAR, whereas CAR needs a modifier that builds closed ABox repairs, and k-lazy needs splitting into k-lazy repairs).

We can note that the "natural instantiation of expansion" considered in [Baget *et al.* 2016], positive closure, that amounts to forgetting the negative inclusions to expand the ABox, differs from that proposed by [Lembo *et al.* 2010] for defining the CAR and ICAR semantics and leads to different semantics (for instance if $\mathcal{T} = \{\exists R \sqsubseteq \neg \exists R^-, \exists R \sqsubseteq A\}$ and $\mathcal{A} = \{R(a,a)\}$, $A(a)$ is not entailed under the CAR and ICAR semantics but is entailed under $\langle RC, \cap \rangle$). This shows that defining how to expand an inconsistent ABox is not completely clear: CAR and ICAR semantics avoid deriving facts that have only inconsistent reasons to hold but do derive conjunctive queries from inconsistent sets of ABox assertions.

2.2.4 Summary

In this section, we reviewed the main inconsistency-tolerant semantics that have been proposed in the literature for querying DL KBs. Some other inconsistency-tolerant semantics based on preferences over assertions or repairs are presented in Chapter 6, and some related works in the arena of database or existential rules are reviewed in Chapter 7. Figure 2.3 summarizes the relationships between inconsistency-tolerant semantics presented in this section, and Table 2.5 the complexity results for these semantics.

In this thesis, the main focus is on the AR semantics which it is the most widely accepted semantics for querying inconsistent data. We also consider the IAR and brave semantics which correspond to the most and less safe inference strategies over repairs and provide natural lower and upper bounds to AR. These two semantics are also interesting by themselves since they have a clear meaning and it may be important in some applications to know the surest answers that can be retrieved from the knowledge base, or all answers that have some consistent reason to hold, when only very reliable answers should be taken into account, or when missing a possible answer has to be avoided.

Example 2.2.36 (Example 2.2.1 cont'd)**.** We conclude our example by summing up for each KB $\langle \mathcal{T}, \mathcal{A}_x \rangle$ under which semantics holds $q(x)$.

	IAR	**ICR**	k**-support**	**AR**	k**-defeater**	**brave**	**ICAR**	**CAR**	k**-lazy**
ann	✓	✓	$k \geq 1$	✓	k	✓	✓	✓	$k \geq 0$
bob		✓	$k \geq 3$	✓	k	✓	✓	✓	$k \geq 2$
$carl$			$k \geq 2$	✓	k	✓		✓	$k \geq 2$
dan					$k \leq 1$	✓			
eva					$k = 0$	✓		✓	$k = 1$
$fred$							✓	✓	

◁

Fig. 2.3 Relationships between inconsistency-tolerant semantics. An arrow from S to S' indicates that if $\mathcal{K} \models_S q$ then $\mathcal{K} \models_{S'} q$.

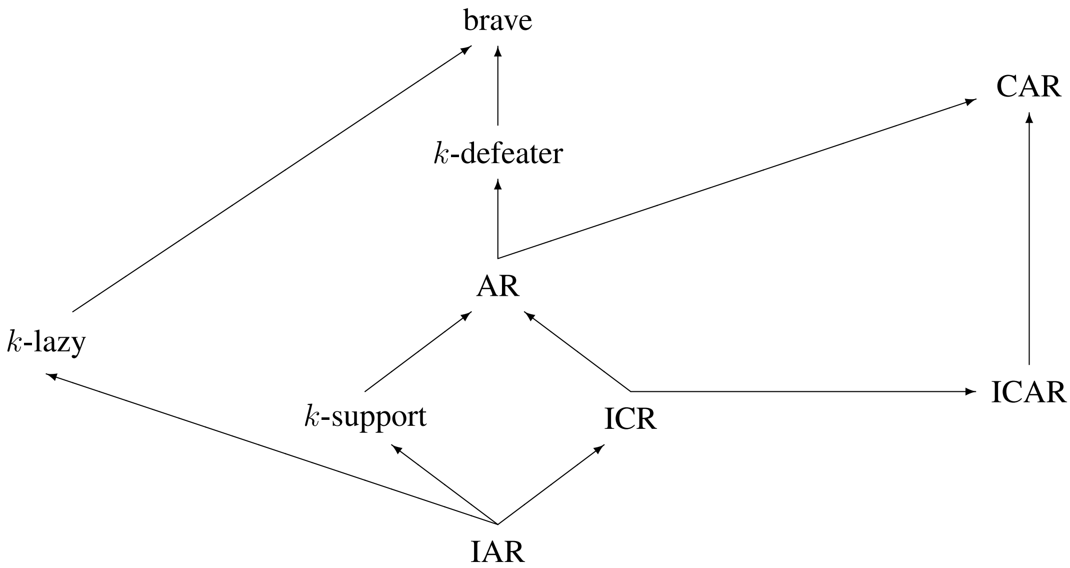


Table 2.5 Data and combined complexity of conjunctive query answering and instance checking over DL-Lite KBs under inconsistency-tolerant semantics.

	Data complexity		**Combined complexity**	
	CQs	**instance queries**	**CQs**	**instance queries**
AR	coNP-co	coNP-co	Π_2^p-co	coNP-co
IAR	in AC^0	in AC^0	NP-co	in P
brave	in AC^0	in AC^0	NP-co	in P
k**-support**	in AC^0	in AC^0	NP-co	in P
k**-defeater**	in AC^0	in AC^0	NP-co	in P
ICR	coNP-co	coNP-co	$\Delta_2^p[O(\log n)]$-co	coNP-co
CAR	coNP-co	in AC^0	Π_2^p-co	in P
ICAR	in AC^0	in AC^0	NP-co	in P
k**-lazy** ($k \geq 1$)	coNP-co	in P	Π_2^p-co	in P

3

EFFICIENT INCONSISTENCY-TOLERANT QUERY ANSWERING IN DL-LITE

In this chapter, we present the algorithms we implemented in our CQAPri prototype system for query answering under AR, IAR and brave semantics over DL-Lite knowledge bases. Then, we describe the experimental setting we built to evaluate it, as well as the results we obtained. Finally, we discuss other existing systems for inconsistency-tolerant query answering.

3.1 Algorithms

We compute the answers that hold under AR, IAR and brave semantics using the basic algorithms ComputeConflicts and ComputeCauses presented in the preceding section and exploiting the properties of such answers.

Brave To decide if a Boolean query is entailed under brave semantics, we simply need to compute its causes with Algorithm 2.3 (ComputeCauses) and verify that the output is not empty.

IAR Since a query holds under IAR semantics just in the case that one of its cause is such that none of its assertions is involved in some conflict, deciding if a query is entailed under IAR semantics can be done by computing its causes with Algorithm 2.3 (ComputeCauses) and the conflicts of the KB with Algorithm 2.2 (ComputeConflicts).

AR For the coNP-complete problem of deciding if a query is entailed under AR semantics, we encode it as a (UN)SAT problem. Figure 3.1 presents the encoding we use, where variables represent assertions of the ABox, and that is satisfiable just in the case that the query does not hold under AR semantics. Intuitively, the assertions that correspond to the variables assigned to true in a truth assignment that satisfies the formula form a consistent subset of the ABox that contains at least one assertion of the conflicts of each cause of the

Fig. 3.1 SAT encoding for AR entailment.

$$\varphi_{\neg q} = \bigwedge_{\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})} \quad \bigvee_{\beta \in \mathsf{confl}(\mathcal{C},\mathcal{K})} x_\beta$$

$$\varphi_{cons} = \bigwedge_{x_\alpha, x_\beta \in \mathsf{vars}(\varphi_{\neg q}), \beta \in \mathsf{confl}(\{\alpha\},\mathcal{K})} \neg x_\alpha \vee \neg x_\beta$$

with $\mathsf{vars}(\varphi_{\neg q})$ the set of variables appearing in $\varphi_{\neg q}$.

query (cf. Definition 2.1.25 for conflicts of a cause). Since this subset is consistent, it can be extended to a repair and such a repair cannot contain any cause for the query, otherwise it would contain a conflict and be inconsistent. The interest of this encoding is that it has as many variables as the number of assertions which are in a conflict with some assertions of a cause of the query. Thus, in practical cases, the size of the formula may be much smaller than the size of the ABox.

Theorem 3.1.1. *Let $\mathcal{K}$ be a DL-Lite$_\mathcal{R}$ KB and q be a Boolean conjunctive query. $\mathcal{K} \not\models_{AR} q$ if and only if $\varphi_{\neg q} \wedge \varphi_{cons}$ is satisfiable, where $\varphi_{\neg q}$ and φ_{cons} are defined in Figure 3.1.*

Proof. Let ν be a truth assignment of the variables of $\varphi_{\neg q} \wedge \varphi_{cons}$ and $\mathcal{R}_\nu = \{\beta \mid \nu(x_\beta) = \mathsf{true}\}$. The formula $\varphi_{\neg q}$ evaluates to true in ν if and only if for every cause $\mathcal{C}$ of q, there exists an assertion β which is in conflict with some assertion of $\mathcal{C}$ and such that $\nu(x_\beta) = \mathsf{true}$, so is in $\mathcal{R}_\nu$. The formula φ_{cons} evaluates to true in ν if and only if there is no α, β which are in a conflict and such that $\nu(x_\alpha) = \mathsf{true}$ and $\nu(x_\beta) = \mathsf{true}$, so if and only if $\mathcal{R}_\nu$ does not contain any conflict of $\mathcal{K}$, i.e. is $\mathcal{T}$-consistent. Hence $\varphi_{\neg q} \wedge \varphi_{cons}$ evaluates to true in ν if and only if $\mathcal{R}_\nu$ is a consistent subset of $\mathcal{A}$ that contradicts each cause for q, thus can be extended to a repair of $\mathcal{K}$ which does not contain any cause for q, since adding a cause for q to $\mathcal{R}_\nu$ would make it inconsistent. We conclude that $\varphi_{\neg q} \wedge \varphi_{cons}$ is satisfiable if and only if there exists a repair of $\mathcal{K}$ that does contain any cause for q, so if and only if $\mathcal{K} \not\models_{\text{AR}} q$. $\square$

Example 3.1.2. We illustrate the encoding on the following KB:

$$\begin{aligned}\mathcal{T} = &\{\mathsf{AProf} \sqsubseteq \mathsf{Prof}, \mathsf{FProf} \sqsubseteq \mathsf{Prof}, \mathsf{Prof} \sqsubseteq \mathsf{PhD}, \mathsf{Postdoc} \sqsubseteq \mathsf{PhD},\\ &\mathsf{AProf} \sqsubseteq \neg\mathsf{FProf}, \mathsf{Prof} \sqsubseteq \neg\mathsf{Postdoc}\}\\ \mathcal{A} = &\{\mathsf{AProf}(a), \mathsf{FProf}(a), \mathsf{Postdoc}(a)\}\end{aligned}$$

The conflicts of $\langle \mathcal{T}, \mathcal{A} \rangle$ are $\{\mathsf{AProf}(a), \mathsf{FProf}(a)\}$, $\{\mathsf{AProf}(a), \mathsf{Postdoc}(a)\}$, and $\{\mathsf{FProf}(a), \mathsf{Postdoc}(a)\}$. We have therefore:

$$\begin{aligned}\mathsf{confl}(\{\mathsf{AProf}(a)\}, \mathcal{K}) = &\{\mathsf{FProf}(a), \mathsf{Postdoc}(a)\}\\ \mathsf{confl}(\{\mathsf{FProf}(a)\}, \mathcal{K}) = &\{\mathsf{AProf}(a), \mathsf{Postdoc}(a)\}\\ \mathsf{confl}(\{\mathsf{Postdoc}(a)\}, \mathcal{K}) = &\{\mathsf{AProf}(a), \mathsf{FProf}(a)\}\end{aligned}$$

For the query $q = \mathsf{Prof}(a)$, which has two causes $\{\mathsf{AProf}(a)\}$ and $\{\mathsf{FProf}(a)\}$, we obtain:

$$\begin{aligned}\varphi_{\neg q} =& (x_{\mathsf{FProf}(a)} \vee x_{\mathsf{Postdoc}(a)}) \wedge (x_{\mathsf{AProf}(a)} \vee x_{\mathsf{Postdoc}(a)})\\ \varphi_{cons} =& (\neg x_{\mathsf{FProf}(a)} \vee \neg x_{\mathsf{Postdoc}(a)}) \wedge (\neg x_{\mathsf{AProf}(a)} \vee \neg x_{\mathsf{Postdoc}(a)}) \wedge (\neg x_{\mathsf{AProf}(a)} \vee \neg x_{\mathsf{FProf}(a)})\end{aligned}$$

A valuation that assigns $x_{\mathsf{Postdoc}(a)}$ to true and $x_{\mathsf{AProf}(a)}$ and $x_{\mathsf{FProf}(a)}$ to false satisfies $\varphi_{\neg q} \wedge \varphi_{cons}$, so $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{\text{AR}} q$.

If we consider now the query $q = \mathsf{PhD}(a)$, which has three causes, $\{\mathsf{AProf}(a)\}$, $\{\mathsf{FProf}(a)\}$, and $\{\mathsf{Postdoc}(a)\}$, we obtain:

$$\begin{aligned}\varphi_{\neg q} =& (x_{\mathsf{FProf}(a)} \vee x_{\mathsf{Postdoc}(a)}) \wedge (x_{\mathsf{AProf}(a)} \vee x_{\mathsf{Postdoc}(a)}) \wedge (x_{\mathsf{AProf}(a)} \vee x_{\mathsf{FProf}(a)})\\ \varphi_{cons} =& (\neg x_{\mathsf{FProf}(a)} \vee \neg x_{\mathsf{Postdoc}(a)}) \wedge (\neg x_{\mathsf{AProf}(a)} \vee \neg x_{\mathsf{Postdoc}(a)}) \wedge (\neg x_{\mathsf{AProf}(a)} \vee \neg x_{\mathsf{FProf}(a)})\end{aligned}$$

The formula $\varphi_{\neg q} \wedge \varphi_{cons}$ is unsatisfiable, so $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{AR}} q$. ◁

Remark 3.1.3. If we allow denial constraints of the form $B_1 \sqcap \ldots \sqcap B_n \sqsubseteq \bot$ in the TBox or of the form of a Boolean CQ ($\exists \vec{y} \psi(\vec{y})$, where ψ is a conjunction of atoms of the forms $A(t)$ or $R(t,t')$, with t,t' individuals or variables from $\vec{y}$) which has to be false, the encoding can be adapted to take into account n-ary conflicts: $\mathcal{K} \not\models_{\text{AR}} q$ if and only if $\varphi^1_{\neg q} \wedge \varphi^2_{\neg q} \wedge \varphi_{cons}$ is satisfiable.

$$\begin{aligned}\varphi^1_{\neg q} &= \bigwedge_{\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})} \bigvee_{\mathcal{B} \in \mathsf{conflicts}(\mathcal{K}), \mathcal{C} \cap \mathcal{B} \neq \emptyset} x_{\mathcal{C},\mathcal{B}}\\ \varphi^2_{\neg q} &= \bigwedge_{x_{\mathcal{C},\mathcal{B}} \in \mathsf{vars}(\varphi^1_{\neg q})} \bigwedge_{\beta \in \mathcal{B} \setminus \mathcal{C}} \neg x_{\mathcal{C},\mathcal{B}} \vee x_\beta\\ \varphi_{cons} &= \bigwedge_{x_\alpha \in \mathsf{vars}(\varphi^2_{\neg q})} \bigwedge_{\mathcal{B} \in \mathsf{conflicts}(\mathcal{K}), \alpha \in \mathcal{B}} \bigvee_{\beta \in \mathcal{B}} \neg x_\beta\end{aligned}$$

When conflicts are not binary, we cannot define the conflicts of a cause as a set of assertions: it may be necessary to use several assertions to contradict a cause. The new variables $x_{\mathcal{C},\mathcal{B}}$ represent each possibility to contradict $\mathcal{C}$ and $\varphi^1_{\neg q}$ expresses that every cause is contradicted. The formula $\varphi^2_{\neg q}$ ensures that when $x_{\mathcal{C},\mathcal{B}}$ is assigned to true, which means that $\mathcal{C}$ is contradicted with the conflict $\mathcal{B}$, every assertion of $\mathcal{B}$ which does not belong to $\mathcal{C}$ is selected, so that adding $\mathcal{C}$ creates a conflict. As in the preceding encoding, φ_{cons} enforces consistency of the subset consisting of the assertions those corresponding variables are assigned to true by preventing all assertions of a conflict to be selected together.

Algorithm 3.4 gives the strongest of these three semantics under which a query holds.

3.2 The CQAPri system

We implemented our query answering framework under AR, IAR and brave semantics over DL-Lite$_\mathcal{R}$ KBs in Java v1.7 within our CQAPri ("Consistent Query Answer-

Algorithm 3.4 ClassifyQuery

Input: a Boolean conjunctive query q, a TBox $\mathcal{T}$, an ABox $\mathcal{A}$
Output: **IAR** if $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{IAR}} q$, **AR** if $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{AR}} q$ and $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{\text{IAR}} q$, **brave** if $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{brave}} q$ and $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{\text{AR}} q$, and **not brave** otherwise

$Conflicts \leftarrow$ ComputeConflicts($\mathcal{T}, \mathcal{A}$)
$ConflictAssertions \leftarrow \bigcup_{\mathcal{B} \in Conflicts} \mathcal{B}$
$Causes \leftarrow$ ComputeCauses($q, \mathcal{T}, \mathcal{A}$)
if $Causes = \emptyset$ **then**
    Output **not brave**
end if
$CausesConfl \leftarrow \emptyset$
for all $\mathcal{C} \in Causes$ **do**
    if $\mathcal{C} \cap ConflictAssertions = \emptyset$ **then**
        Output **IAR**
    else
        $CausesConfl \leftarrow CausesConfl \cup \{\beta \mid \alpha \in \mathcal{C}, \{\alpha, \beta\} \in Conflicts\}$
    end if
end for
$\varphi \leftarrow$ ConstructEncoding($CausesConfl, Conflicts$)
if φ is unsatisfiable **then**
    Output **AR**
else
    Output **brave**
end if

where constructEncoding constructs the SAT encoding of Theorem 3.1 from the conflicts of the causes for q and conflicts of $\mathcal{K}$.

ing with Priorities") tool. CQAPri is built on top of the relational database server PostgreSQL v9.3.2 (www.postgresql.org), the Rapid v1.0 query rewriting engine for DL-Lite [Chortaras *et al.* 2011], and the SAT4J v2.3.4 SAT solver [Berre & Parrain 2010]. All these building blocks are used with their default settings.

CQAPri classifies a query answer $\vec{a}$ into one of 3 classes:

- **Possible**: $\mathcal{K} \models_{\text{brave}} q(\vec{a})$ and $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$
- **Likely**: $\mathcal{K} \models_{\text{AR}} q(\vec{a})$ and $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$
- **(Almost) sure**: $\mathcal{K} \models_{\text{IAR}} q(\vec{a})$

To do that, CQAPri follows the line of Algorithm 3.4, but uses consistent images instead of causes to decide if a query holds under one or the other semantics, since every cause is an image and every consistent image contains a cause, and removing non-minimal ones is time consuming (cf. Section 4.3.3, Figure 4.5 for an insight of this cost).

CQAPri handles ABoxes stored in PostgreSQL, while it keeps the TBox in-memory. It computes the set of conflicts for the KB in a preprocessing phase since it is query-independent. Conflicts are computed as in Algorithm 2.2: the SQLized rewritings of the queries looking for counter-examples to the negative TBox inclusions are evaluated over the ABox, their images are retrieved and stored as a graph whose vertices are assertions and edges indicate images, and finally the non-minimal ones are discarded by removing edges between any self-inconsistent assertion and the others to obtain the conflict graph.

When a query arrives, CQAPri evaluates it over the ABox using its SQLized rewriting, to obtain its *candidate answers* and their images. Candidate answers define a superset of the answers holding under the brave, AR and IAR semantics. Among the candidate answers, CQAPri identifies the IAR ones, by checking whether there is some image whose assertions have no outgoing edges in the conflict graph, since such an image contains a cause such that none of its assertions is involved in a conflict. It also identifies those which are not brave-answers by discarding the inconsistent images, that contain an edge of the conflict graph: an answer that has only such images does not hold under brave semantics. Finally, for brave-answers that are not found to be IAR-answers, deciding whether they are entailed under the AR semantics is done using the SAT encodings from the preceding section. Using consistent images instead of causes is not a problem here because the set of causes is included in the set of images and every consistent image contains a cause, so it is possible to consistently contradict every cause iff it is possible to consistently contradict every consistent image.

3.3 Experiments

We conducted experiments to empirically study the properties of our framework. We study in particular the impact of the data quality and size on CQAPri behavior, and the proportion of answers in the different classes.

3.3.1 The CQAPri benchmark

To evaluate CQAPri, we needed a DL-Lite$_\mathcal{R}$ KB with a large and inconsistent ABox. Very few experiments have been done on inconsistent KBs, and the only DL-Lite$_\mathcal{R}$ benchmark we found was not suitable for our case (cf. Section 3.4.2), so we designed our own.

Ontology

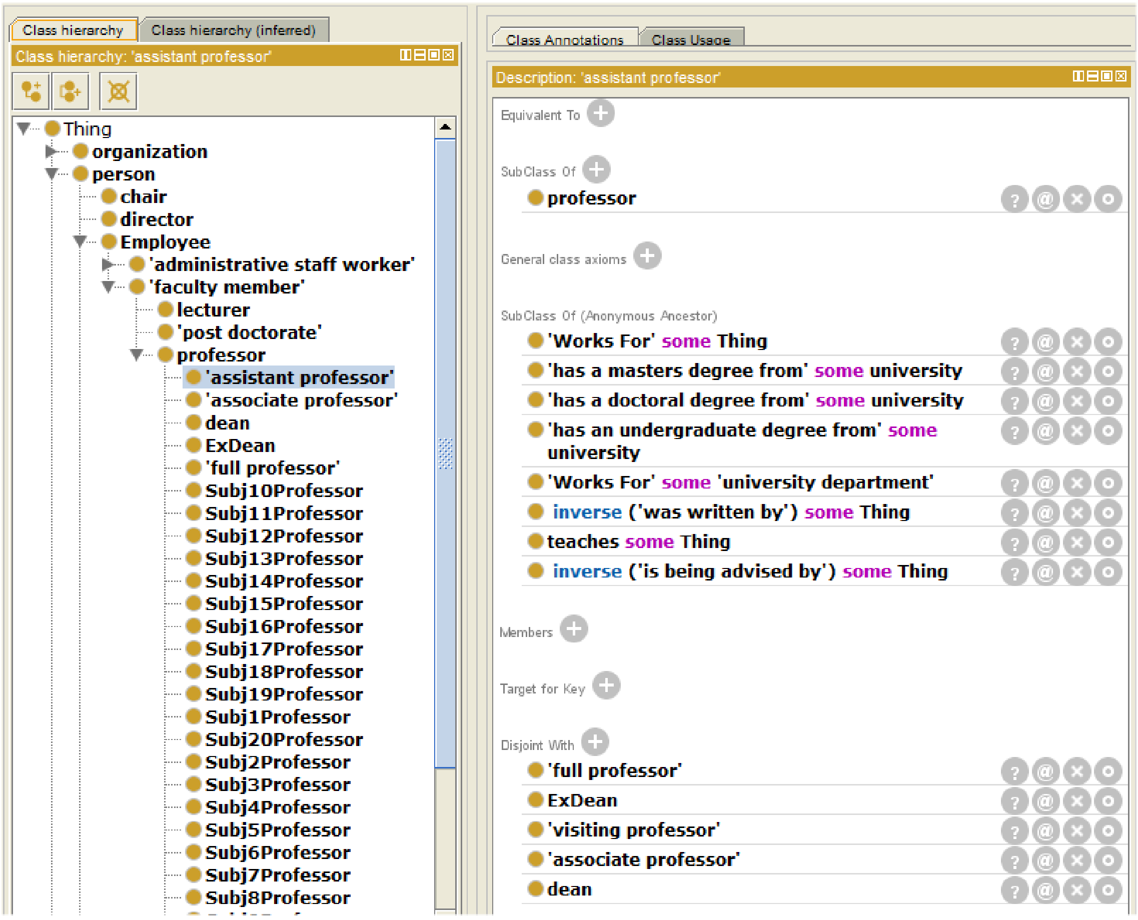
We built our TBox over the $\text{LUBM}^{\exists}_{20}$ TBox from [Lutz *et al.* 2013], which provides an extended DL-Lite$_\mathcal{R}$ version of the well-known $\mathcal{ELI}$ TBox of the Lehigh University Benchmark (LUBM) of [Guo *et al.* 2005] that describes the university domain. $\text{LUBM}^{\exists}_{20}$ differs from the original LUBM by the removal of the axioms that go beyond DL-Lite$_\mathcal{R}$ but also by the addition of concept inclusions, many of which having existential restrictions on the right-hand side, and of subconcepts to increase the size of the ontology. $\text{LUBM}^{\exists}_{20}$ comprises 127 concepts, 27 roles and 202 positive inclusions. To allow for contradiction, we added negative inclusions to state the disjointness of pairs of concepts or roles having the same closest super-concept or super-role. Such concepts and roles appear at the same level in the TBox (that is, have the same distance to the top concept Thing). We excluded a small number of such inclusions when they did not seem to reflect the intended meaning of the concepts or roles. Figure 3.2 illustrates how we added negative inclusions for an example concept AssistantProfessor. We added 875 negative inclusions in total. This apparently huge number of constraints results from the many pairwise disjoint concepts or roles used in the TBox.

Datasets

We generated ABoxes of increasing sizes with the Extended University Data Generator (EUGen) provided with $\text{LUBM}^{\exists}_{20}$ by setting its *data completeness* parameter (i.e. the percentage of individuals from a given concept for which roles describing this concept are indeed filled) to its default value of 95%, which seems realistic from the application viewpoint. All the generated ABoxes were found consistent w.r.t. our enriched TBox, meaning that the added disjointness constraints were faithful to the reused benchmark. The size of these ABoxes ranges from 75,663 to 9,938,139 assertions, which corresponds to 1 to 100 universities in EUGen settings, and each ABox is included in the larger ones: the smallest one corresponds to university 0, the largest one to universities 0 to 99.

Inconsistencies were introduced by reviewing all the assertions of the consistent ABox, and contradicting the presence of an individual in a concept assertion with probability p, and the presence of each individual in a role assertion with probability $p/2$. A contradicting assertion is built by stating that the considered individual also belongs to a disjoint but close concept, i.e. the two concepts have the same closest super-concept which is not the top concept Thing. Note that a concept may here be an unqualified existential role restriction. The contradicting assertion is added either explicitly or implicitly by choosing one of its specializations (obtained by query rewriting). The concept or role that is used to build the assertion which will actually be added to the ABox is chosen among all rewritings of all possible contradicting assertions with a uniform probability distribution. We chose to

Fig. 3.2 The left side displays a part of the concept hierarchy of the LUBM$^{\exists}_{20}$ ontology, the right upper part shows of which concepts AssistantProfessor is a subconcept (here the single concept Professor). The right lower part displays the negative inclusions added between AssistantProfessor and concepts with the same closest super-concept Professor: FullProfessor, ExDean, VisitingProfessor, AssociateProfessor and Dean. We did not add disjointness axioms with the concepts SubjXProfessor, because such concepts indicate the domain of a professor, which is independent from its seniority.



generate such inconsistencies because they seem quite natural in real applications (e.g. using by mistake AssistantProfessor in place of AssociateProfessor). Conflicting assertions thus introduced are in turn processed as described above to create a few more complex conflicts. Additionally, for every role assertion, its individuals are switched with probability $p/10$. We chose to generate such misuses of roles because they also seem quite natural mistakes and may lead to inconsistencies (e.g. inverting the Faculty and Course in a TeacherOf role assertion).

Example 3.3.1. Below are four assertions that have been created to conflict some assertions of the original consistent ABoxes.

- The assertion Subj4Course($Department11.University1/GraduateCourse33$) contradicts Subj20Course($Department11.University1/GraduateCourse33$) because these concepts are disjoint (Subj20Course $\sqsubseteq \neg$Subj4Course).
- The assertion Subj3Department($University462$) has been inserted to contradict MastersDegreeFrom($Department22.University0/Lecturer2, University462$): indeed, the range of MastersDegreeFrom is University ($\exists$MastersDegreeFrom$^-$ $\sqsubseteq$ University), which is disjoint with Department (University $\sqsubseteq \neg$Department), which has Subj3Department has subconcept (Subj3Department $\sqsubseteq$ Department}).
- PublicationResearch($DUMMY\_1\_1\_749, Department18.University1/Course32$) conflicts TakesCourse($Department18.University1/UndergraduateStudent124$, $Department18.University1/Course32$). Indeed, TakesCourse has Course for range, which is disjoint with Research which is the range of PublicationResearch.
- Finally MemberOf($Department5.University1, Department5.University1/$ $UndergraduateStudent47$) is obtained by switching the two individuals of an assertion. Note that this kind of assertion may induce that an individual belongs to a totally different concept, since the domain and the role of a concept have often no common super-concept other than Thing. Such inversions generally yield a lot of conflicts.

$\triangleleft$

For each of the 100 universities that constitute our consistent ABoxes, we set $p = 0.002$ and generate 50 batches of conflicting assertions to insert, using the method described above. We obtain inconsistent ABoxes with growing ratios of assertions involved in some conflict by adding the n first batches of conflicting assertions to each university of the original consistent ABox, n ranging from 1 to 50, that roughly leads to a percentage of assertions involved in some conflict varying from about 3% to about 46%. We consider that ABoxes with a few percent of assertions in conflicts are realistic, but we also built ABoxes with a huge number of conflicts in order to study the impact of the data quality on the efficiency of our approach.

Table 3.1 displays the characteristics in terms of size, inserted assertions, and number and percentage of assertions involved in conflicts of the ABoxes of our benchmark. Every ABox's id uXcY indicates the number X of universities generated by EUGen and the number of queries batches used to add conflicts Y. Note that our method ensures that uXcY $\subseteq$ uX$'$cY when X $\leq$ X$'$ and uXcY $\subseteq$ uXcY$'$ when Y $\leq$ Y$'$.

ABoxes are stored as relational databases in PostgreSQL with one table per concept and role, of one or two columns (named s for concept and s and o for role) that contains the individuals involved in that concept or role. B-Tree indexes have been created for each table, on s for concepts and (s, o) and (o, s) for roles. The concepts, roles and individuals names are encoded by integers, and a dictionary table relates each name to its identifier.

Queries

Figure 3.3 displays the 20 queries used in our experiments and Table 3.2 summarizes their characteristics. They have between 1 and 8 atoms, with an average of 4.25 atoms. Their rewritings produced with Rapid have between 2 and 202,710 CQs, 23,185.95 on average.

Queries q14 to q20 were available on websites associated with [Pérez-Urbina *et al.* 2009, Lutz *et al.* 2013, Rosati *et al.* 2012], and we designed the others ourselves. We chose queries that use some concepts that have disjoint specializations to get more chance to get answers that hold under AR semantics and not under IAR semantics.

3.3.2 Experimental setting

All experiments reported in this thesis were run on an Intel Xeon X5647 at 2.93 GHz with 16 GB of RAM, running CentOS 6.8. Reported times are averaged over 5 runs.

We did not measure the time that our prototype takes to present the results (i.e. translating the answers back in the original terminology with the dictionary table of the database and printing the results in text files), since the goal our experiments was to study the properties of the computation of query answers rather than their presentation, that a real OMQA system would probably handles in a more refined and efficient way.

3.3.3 Experimental results

We empirically study the properties of our consistent query answering framework. We measure the time spent in the different phases of query answering under IAR, AR and brave semantics, and how it varies with the size of the ABoxes and the ratio of assertions involved in some conflicts. We also consider the evolution of the number of answers under each semantics.

The time it took CQAPri to be up and ready to answer queries is dominated by the construction of the conflict graph for the ABox (Table 3.3); it took about 2 seconds to load the TBox, construct the queries that correspond to the violation of negative inclusions, and open the PostgreSQL connection to the ABox. The time for the construction of the conflict graph has a linear behavior w.r.t. the size of the ABox, and the more conflicts there are, the higher is the slope.

Table 3.4 shows, per ABox and query pair, how many Sure, Likely and Possible answers were identified among the candidate answers. In this table, OOM means that CQAPri ran out of memory. In all our experiments, we found only a few candidate answers that are not brave: only q9 got such answers, on all ABoxes (up to 802 inconsistent answers on u100conf50). Only 9 queries out of 20 got Likely answers over some ABoxes, 5 of them have only one

Table 3.1 Characteristics of ABoxes used in experiments.

ABox id	size	% assertions added to uXc0	# assertions in some conflict	% assertions in some conflict
u1c0	75,663	0	0	0
u1c1	75,724	0.08	2,373	3
u1c5	75,951	0.38	6,412	8
u1c10	76,201	0.70	10,891	14
u1c20	76,821	1.51	20,175	26
u1c30	77,447	2.30	26,086	34
u1c50	78,593	3.73	34,814	44
u5c0	463,325	0	0	0
u5c1	463,691	0.08	12,191	3
u5c5	465,157	0.39	45,906	10
u5c10	466,919	0.77	83,263	18
u5c20	470,674	1.56	137,836	29
u5c30	474,368	2.33	172,245	36
u5c50	481,400	3.75	221,900	46
u20c0	1,981,872	0	0	0
u20c1	1,983,493	0.08	69,597	4
u20c5	1,989,788	0.40	253,141	13
u20c10	1,997,445	0.78	408,398	20
u20c20	2,013,048	1.55	610,271	30
u20c30	2,028,069	2.28	748,664	37
u20c50	2,056,957	3.65	946,819	46
u50c0	4,934,691	0	0	0
u50c1	4,938,737	0.08	224,131	5
u50c5	4,954,494	0.40	686,159	14
u50c10	4,973,292	0.78	1,034,226	21
u50c20	5,010,776	1.52	1,517,499	30
u50c30	5,046,802	2.22	1,865,679	37
u50c50	5,115,473	3.53	2,353,739	46
u100c0	9,938,139	0	0	0
u100c1	9,946,144	0.08	546,708	5
u100c5	9,977,656	0.40	1,381,298	14
u100c10	10,014,894	0.77	2,077,201	21
u100c20	10,087,801	1.48	3,069,321	30
u100c30	10,157,192	2.16	3,755,732	37
u100c50	10,289,863	3.42	4,728,588	46

Fig. 3.3 Queries used in experiments.

$$
\begin{aligned}
\text{q1} =& \text{Person}(x) \wedge \text{takesCourse}(x, y)\\
\text{q2} =& \text{Employee}(x) \wedge \text{publicationAuthor}(y, x)\\
\text{q3} =& \text{Professor}(x) \wedge \text{teacherOf}(x, y) \wedge \text{worksFor}(x, Department0.University0)\\
\text{q4} =& \text{FullProfessor}(x) \wedge \text{publiAuthor}(y, x) \wedge \text{teacherOf}(x, z) \wedge \text{advisor}(u, x) \wedge\\
& \text{graduateStudent}(u) \wedge \text{degreeFrom}(x, v) \wedge \text{degreeFrom}(u, w)\\
\text{q5} =& \exists y \text{Person}(Department2.University0/Graduatudent131) \wedge\\
& \text{takesCourse}(Department2.University0/GraduateStudent131, y) \wedge\\
& \text{GraduateCourse}(y) \wedge \text{takesCourse}(x, y) \wedge \text{Person}(x)\\
\text{q6} =& \exists y \text{Employee}(x) \wedge \text{publicationAuthor}(y, x) \wedge \text{Employee}(z) \wedge \text{publicationAuthor}(y, z) \wedge\\
& \text{memberOf}(x, Department4.University0) \wedge\\
& \text{memberOf}(z, Department4.University0)\\
\text{q7} =& \exists x \text{Employee}(x) \wedge \text{memberOf}(x, Department2.University0) \wedge \text{degreeFrom}(x, y)\\
\text{q8} =& \exists y \text{teacherOf}(x, y) \wedge \text{degreeFrom}(x, University532)\\
\text{q9} =& \exists y u \text{Employee}(x) \wedge \text{memberOf}(x, u) \wedge \text{degreeFrom}(x, y) \wedge \text{Employee}(z) \wedge\\
& \text{memberOf}(z, u) \wedge \text{degreeFrom}(z, y)\\
\text{q10} =& \exists u \text{Employee}(x) \wedge \text{memberOf}(x, u) \wedge \text{degreeFrom}(x, University532) \wedge \text{Employee}(z) \wedge\\
& \text{memberOf}(z, u) \wedge \text{degreeFrom}(z, University532)\\
\text{q11} =& \exists y \text{Faculty}(x) \wedge \text{publicationAuthor}(y, x)\\
\text{q12} =& \text{Organization}(x)\\
\text{q13} =& \text{Employee}(x)\\
\text{q14} =& \text{Student}(x) \wedge \text{advisor}(x, y) \wedge \text{Faculty}(y) \wedge \text{takesCourse}(x, z) \wedge \text{teacherOf}(y, z) \wedge\\
& \text{Course}(z)\\
\text{q15} =& \text{Person}(x) \wedge \text{worksFor}(x, y) \wedge \text{Organization}(y)\\
\text{q16} =& \exists z u \text{Student}(x) \wedge \text{takesCourse}(x, y) \wedge \text{Course}(y) \wedge \text{teacherOf}(z, y) \wedge \text{Faculty}(z) \wedge\\
& \text{worksFor}(z, u) \wedge \text{Department}(u) \wedge \text{memberOf}(x, u)\\
\text{q17} =& \exists y z \text{Faculty}(x) \wedge \text{degreeFrom}(x, y) \wedge \text{University}(y) \wedge \text{subOrganizationOf}(z, y) \wedge\\
& \text{Department}(z) \wedge \text{memberOf}(x, z)\\
\text{q18} =& \exists y z \text{Publication}(x) \wedge \text{publicationAuthor}(x, y) \wedge \text{Professor}(y) \wedge\\
& \text{publicationAuthor}(x, z) \wedge \text{Student}(z)\\
\text{q19} =& \exists z u \text{University}(x) \wedge \text{University}(y) \wedge \text{memberOf}(z, x) \wedge \text{Student}(z) \wedge \text{University}(y) \wedge\\
& \text{memberOf}(u, y) \wedge \text{Professor}(u) \wedge \text{advisor}(z, u)\\
\text{q20} =& \text{takesCourse}(x, y) \wedge \text{Student}(x) \wedge\\
& \text{teacherOf}(Department0.University0/AssociateProfessor0, y) \wedge \text{Course}(y)
\end{aligned}
$$

Table 3.2 Queries in terms of shape, numbers of atoms, variables, constants, rewritings, and rewriting time (Rapid).

id	shape	#atoms	#variables	#constants	#rewritings	rewriting time (ms)
q1	chain	2	2	0	80	4
q2	chain	2	2	0	44	3
q3	tree	3	2	1	58	4
q4	dag	7	6	0	25	3
q5	dag	5	2	1	6,401	88
q6	dag	5	3	1	8,240	742
q7	tree	3	2	1	450	7
q8	tree	2	2	1	155	4
q9	dag	6	4	0	202,579	15,917
q10	dag	6	3	1	202,710	33,865
q11	chain	2	2	0	35	3
q12	atomic	1	1	0	44	3
q13	atomic	1	1	0	44	3
q14	dag	6	3	0	23	3
q15	chain	3	2	0	2	3
q16	dag	8	4	0	3,887	124
q17	chain	6	3	0	14,700	190
q18	tree	5	3	0	667	13
q19	dag	8	4	0	23,552	920
q20	chain	4	2	1	23	3

Table 3.3 Construction of conflict graph in milliseconds w.r.t. size uX and conflicts cY.

	c1	c5	c10	c20	c30	c50
u1	2,033	2,384	2,498	2,710	2,920	3,113
u5	7,758	8,542	8,920	9,644	10,462	11,230
u20	27,748	29,878	31,792	34,683	36,982	40,586
u50	73,031	78,673	80,563	91,122	100,134	118,375
u100	153,476	166,234	177,441	200,468	221,172	247,191

or two atoms (q1, q2, q11, q12, q13) while the others are more complex (q5, q6, q9, q18). Such answers show up as these queries are general and involve concepts with many disjoint sub-concepts. For 67.5% of our ABox and query pairs, AR does not provide any additional answers compared to IAR. However, in q5 and uXc20 cases, all answers are AR and not IAR. For such selective queries, using the AR semantics rather than IAR may be necessary to get answers. Unsurprisingly, for a given ABox size, when the proportion of conflicting assertions increases, the number of Sure answers decreases while the number of Likely and Possible answers increases. This incurs a higher computational cost since a call to the SAT solver is needed for each non-IAR-answer to decide if it holds under AR semantics or not.

Figure 3.4 shows the evolution of the time spent by CQAPri for AR query answering w.r.t. the size of the ABox, when the proportion of conflicting assertions is a few percent, as it is likely to be in most real applications, about 30%, and about 45%. Figure 3.5 shows the evolution of the time spent by CQAPri for AR query answering w.r.t. the proportion of ABox assertions involved in some conflicts, for small, intermediate and big ABoxes. Figure 3.6 shows the proportion of the query answering time spent in rewriting the query, executing the rewritten query to get candidate answers, filtering the IAR- or not brave-answers, and identifying the AR-answers among the remaining answers.

At first sight, there are two outlier queries q4 and q9 whose answering times have an exponential-like growth w.r.t. ABox size even in realistic cases. Query q4 is very sensitive both to ABox size and ratio of conflicts, while q9 is rather robust to conflicts. This comes from the uncommon characteristics of these queries. Indeed, q9 has 202,579 rewritings, with 4 variables and no constant, that leads to a very costly execution over the database, as illustrated on Figure 3.6 where almost 90% of the time is spent in executing the rewritten query even in the case of the largest ABox with the highest percentage of conflicts. Query q10 that differs from q9 only by the introduction of a constant, behaves very differently since its answering time stabilizes quickly. Regarding q4, it has a high number of atoms and variables, which are all free, that yields a huge number of answers (10,362,220 answers on u100c10), that becomes quickly non-IAR since their causes involve lots of assertions. These two queries are interesting to challenge our system but are not realistic, especially q4.

For the other queries, CQAPri scales up to large ABoxes when the proportion of assertions involved in some conflict is only a few percent, and even a few tens percent for most of the queries. The increase of the number of non-IAR-answers when the proportion of conflicts increases generally significantly augments the time spent in this last phase. It explains that

Table 3.4 Number of answers in the different classes for a growing proportion of conflicts and three sizes of ABoxes.

	u1c1			u1c5			u1c10			u1c20			u1c30			u1c50		
	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible
q1	20029	0	380	18261	0	2148	16524	0	3885	12538	7	7864	10286	11	10112	6646	19	13747
q2	7215	20	12	7077	57	146	6887	124	308	6284	402	734	5693	652	1174	4728	887	2087
q3	85	0	0	73	0	12	67	0	18	0	0	85	0	0	87	0	0	87
q4	78101	0	5636	62478	0	22159	48745	0	35892	24545	0	60236	12776	0	72221	4806	0	80839
q5	10	0	0	10	0	0	10	0	0	0	10	0	0	10	0	0	0	10
q6	235	0	0	224	0	14	194	9	44	177	14	110	147	39	129	0	0	342
q7	136	0	1	130	0	7	0	0	138	0	0	138	0	0	142	0	0	149
q8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q9	1291	3	80	1220	11	172	1138	26	259	1002	68	406	897	106	530	783	116	741
q10	0	0	3	0	0	3	0	0	6	0	0	6	0	0	6	0	0	7
q11	534	0	4	524	0	20	513	0	38	471	4	89	440	5	140	385	7	236
q12	1180	11	10	1140	49	28	1092	90	57	999	174	117	930	231	204	802	345	350
q13	1069	3	8	1054	11	31	1034	24	56	966	71	122	889	117	204	783	169	351
q14	191	0	4	177	0	18	157	0	38	98	0	97	65	0	130	36	0	159
q15	405	0	102	337	0	171	235	0	273	99	0	409	34	0	474	0	0	515
q16	13545	0	3987	10182	0	7350	6669	0	10863	2052	0	15480	392	0	17140	0	0	17532
q17	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
q18	3107	0	66	2959	0	214	2770	0	403	2302	0	872	1860	0	1319	1319	0	1871
q19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q20	50	0	0	49	0	1	26	0	24	25	0	25	2	0	48	0	0	50

	u20c1			u20c5			u20c10			u20c20			u20c30			u20c50		
	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible
q1	532331	0	12776	485196	11	59900	434068	47	110995	344391	190	200539	279492	395	265246	187504	758	356922
q2	189186	228	1019	184238	1642	5628	177611	3902	11443	163260	9927	22489	150280	15085	32790	127098	23819	52117
q3	85	0	0	73	0	12	67	0	18	0	0	85	0	0	87	0	0	87
q4	1123422	0	944364	115026	0	1956534	8955	0	2067051	118	0	2082472	0	0	2092082	0	0	2114007
q5	10	0	0	10	0	0	10	0	0	0	10	0	0	10	0	0	0	10
q6	235	0	0	224	0	14	194	9	44	177	14	110	147	39	129	0	0	342
q7	91	0	46	35	0	102	0	0	138	0	0	138	0	0	142	0	0	149
q8	0	0	31	0	0	31	0	0	31	0	0	31	0	0	32	0	0	32
q9	33433	60	2714	29222	123	7504	27581	126	9873	25701	59	13282	24150	96	16196	21462	267	21419
q10	0	0	58	0	0	58	0	0	61	0	0	62	0	0	64	0	0	66
q11	14331	0	145	13975	3	708	13514	8	1410	12613	42	2781	11798	96	4066	10329	267	6373
q12	7082	218	251	6460	649	1044	6120	810	1997	5830	880	3881	5646	914	5548	5395	991	8769
q13	28891	64	204	28315	348	1033	27512	779	2054	25791	1780	4028	24209	2658	5938	21471	4185	9430
q14	4785	0	166	4165	0	786	3529	0	1422	2539	0	2412	1853	0	3098	1007	0	3944
q15	12050	0	1702	7877	0	5896	4448	0	9358	1715	0	12143	628	0	13278	54	0	13946
q16	396411	0	72138	234542	0	234007	115090	0	353459	33936	0	434613	9413	0	459136	585	0	467964
q17	27	0	10	3	0	34	0	0	37	0	0	37	0	0	37	0	0	39
q18	81760	0	1342	76118	0	7020	69639	0	13545	58294	0	24959	48975	0	34375	34795	0	48770
q19	0	0	1	0	0	3	0	0	5	0	0	8	0	0	15	0	0	20
q20	50	0	0	49	0	1	26	0	24	25	0	25	2	0	48	0	0	50

	u100c1			u100c5			u100c10			u100c20			u100c30			u100c50		
	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible	Sure	Likely	Possible
q1	2675887	3	65067	2433476	57	307432	2171624	253	569097	1732258	961	1007794	1398917	1870	1340294	934012	3751	1803546
q2	946599	1003	5807	921744	8217	29012	888761	20619	56374	819168	50118	109934	755044	77267	159584	637443	124113	254737
q3	85	0	0	73	0	12	67	0	18	0	0	85	0	0	87	0	0	87
q4	702009	0	9614733	0	0	10339007	0	0	10362220	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
q5	10	0	0	10	0	0	10	0	0	0	10	0	0	10	0	0	0	10
q6	235	0	0	224	0	14	194	9	44	177	14	110	147	39	129	0	0	342
q7	34	0	103	1	0	136	0	0	138	0	0	138	0	0	142	0	0	149
q8	0	0	187	0	0	188	0	0	188	0	0	188	0	0	189	0	0	190
q9	152404	110	27107	140733	18	41939	136664	35	49594	128616	192	64820	121104	451	78963	107220	1300	105450
q10	0	0	293	0	0	294	0	0	299	0	0	310	0	0	319	0	0	326
q11	71756	0	739	69907	14	3627	67723	35	7089	63411	192	13778	59446	451	19956	51791	1300	31764
q12	31955	566	1109	30977	674	5066	30224	713	9572	29074	1166	18162	28256	1733	26118	27002	2849	41644
q13	144313	308	1014	141289	1853	5153	137356	4056	10055	129083	8902	19737	121332	13397	28913	107258	21279	46553
q14	23330	0	777	20271	0	3836	17222	0	6885	12390	0	11717	9083	0	15024	4942	0	19165
q15	61189	0	7584	40044	0	28820	23163	0	45806	7693	0	61492	2494	0	66918	221	0	69599
q16	2029091	0	323720	1182399	0	1170412	594303	0	1758508	150512	0	2202299	37860	0	2314951	2374	0	2350437
q17	28	0	190	0	0	219	0	0	220	0	0	221	0	0	222	0	0	226
q18	406817	0	7330	378618	0	35727	347379	0	67228	291909	0	123116	246077	0	169421	174140	17	242252
q19	0	0	5	0	0	13	0	0	26	0	0	56	0	0	81	0	0	124
q20	50	0	0	49	0	1	26	0	24	25	0	25	2	0	48	0	0	50

Fig. 3.4 Time in seconds for query answering w.r.t. the size of the ABox for three ratios of conflicts (about 4%, 30%, and 45% of assertions involved in some conflict). For readability, the two figures on the right focus on the queries whose answering times are lower and whose behaviors are thus not visible on the first one.

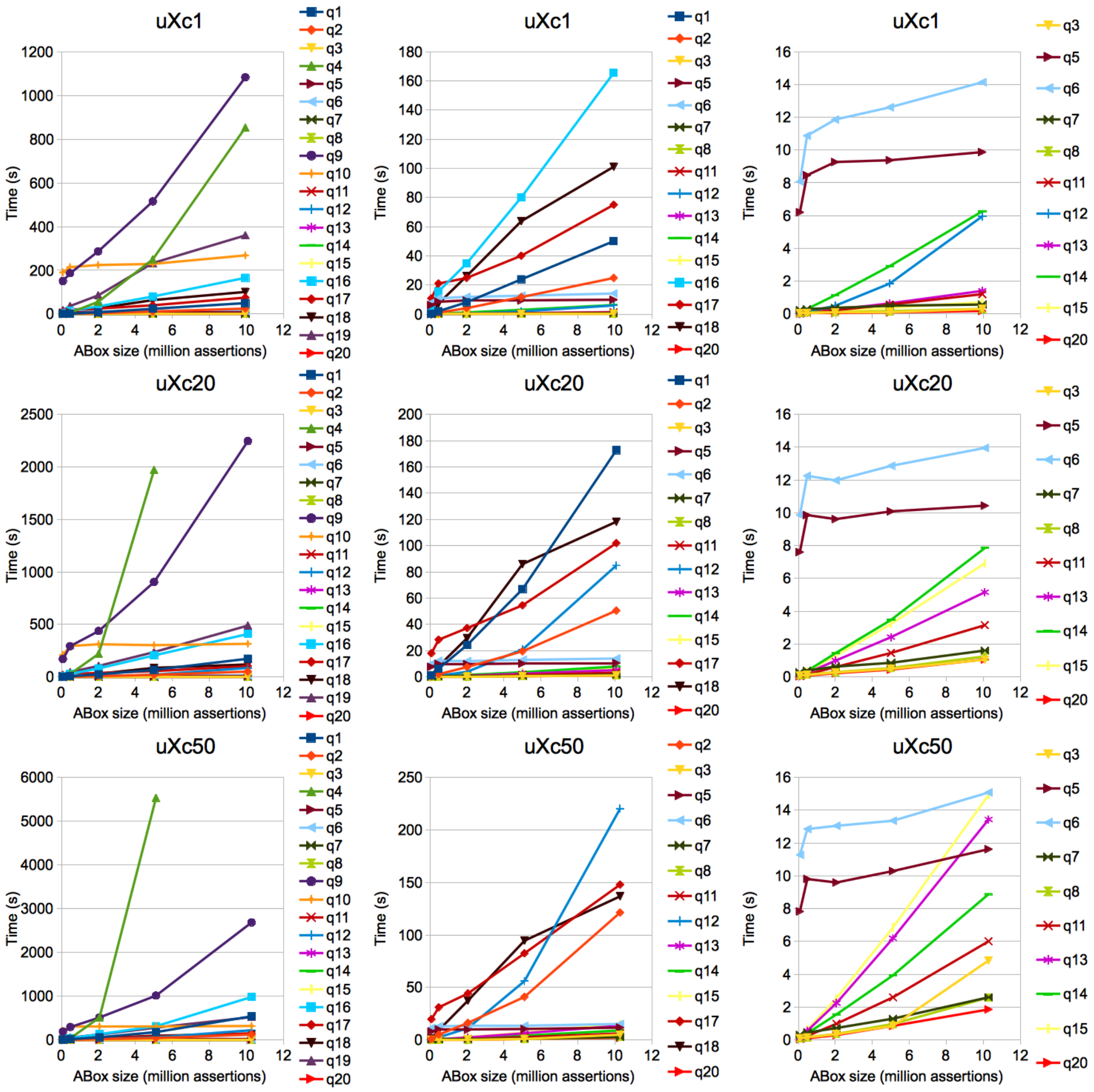


Fig. 3.5 Time in seconds for query answering w.r.t. the ratios of conflicts for three ABox sizes (about 76K, 2 million, and 10 million assertions). For readability, the two figures on the right focus on the queries whose answering times are lower and whose behaviors are thus not visible on the first one.

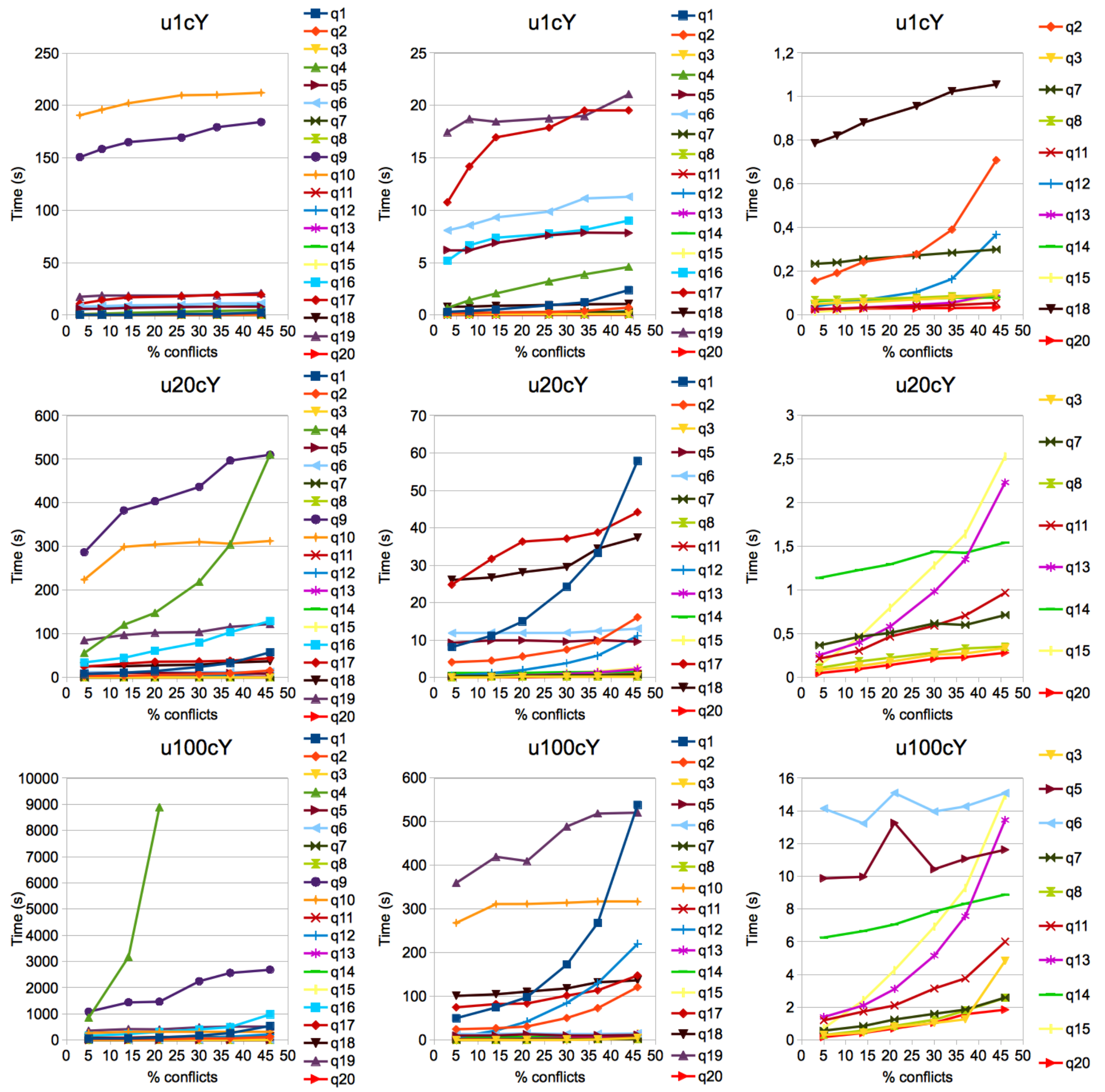
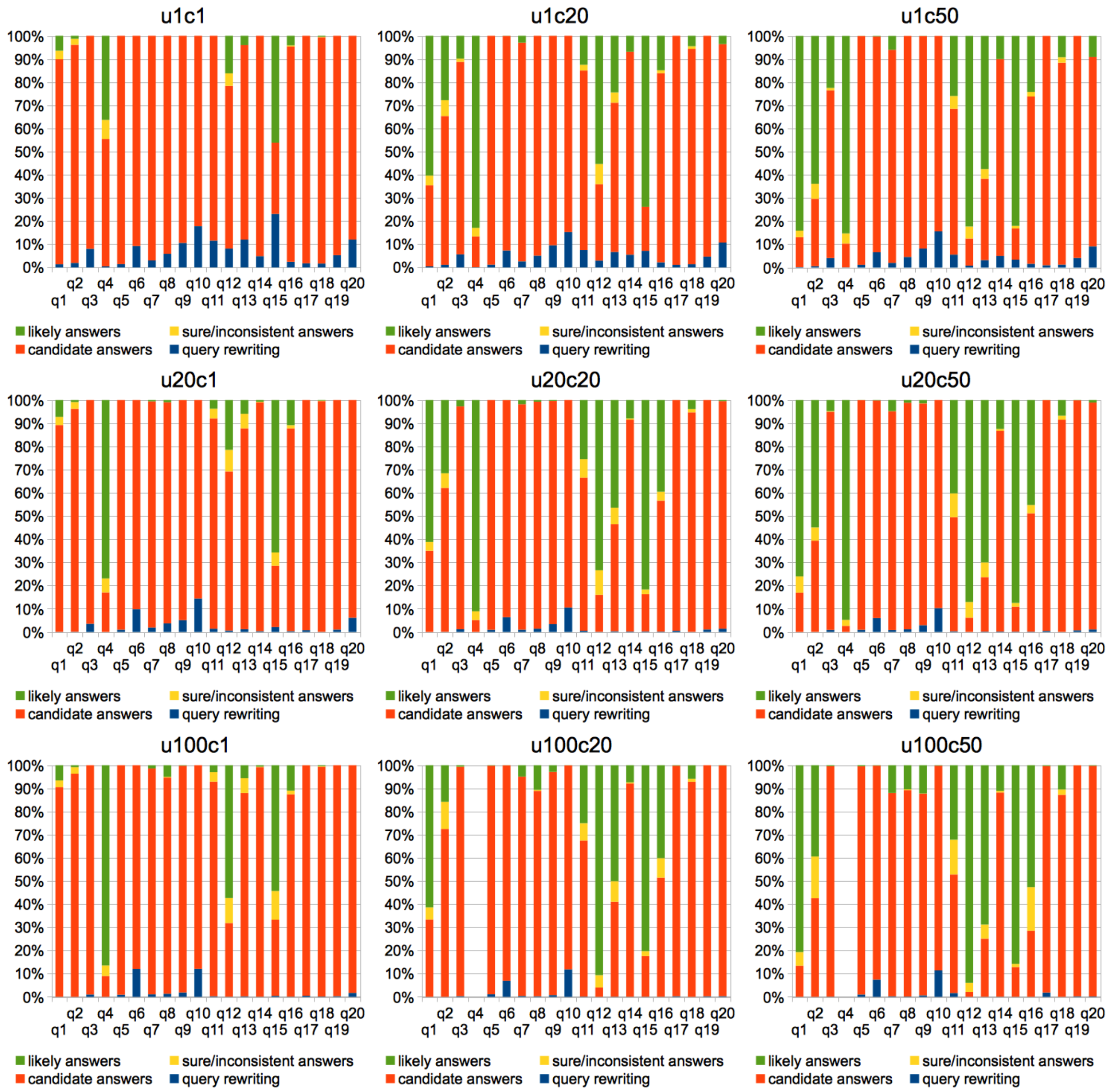


Fig. 3.6 Proportion of time spent by CQAPri in the different phases of query answering on 9 ABoxes: the two lower bars are the time for rewriting the query and executing the rewritten query to get candidate answers, and the two upper bars represent the time needed to classify such answers, by identifying the IAR- and non-brave-answers in a first step, then the AR-answers.



the lower the ratio of conflicts, the more query answering time shows a linear behavior w.r.t. the ABox size.

The average time to classify an answer is generally under the millisecond, and of at most 7 ms (for q19 on u20c50). The average time to decide if a brave and non-IAR-answer is AR or not is generally a little higher but also under the millisecond.

CQAPri scales overall in settings from realistic to more artificial ones. Our experiments thus demonstrate that the AR-answers can be computed in practice and that this is due to the fact that the IAR semantics often constitutes a very good approximation of the AR semantics.

3.4 Discussion: systems and benchmarks for inconsistency-tolerant query answering

3.4.1 Systems for inconsistency-tolerant query answering

In terms of implemented tools and benchmarks for inconsistency-tolerant query answering over DL KBs, we are aware of two systems : the QuID system [Rosati *et al.* 2012, Lembo *et al.* 2015] that handles IAR semantics for CQs and DL-Lite$_{\mathcal{A}}$ KBs, and the system of [Du *et al.* 2013] for querying $\mathcal{SHIQ}$ KBs under a variant of AR semantics with weight on ABox assertions that handles CQs without non-distinguished variables that reduce to assertions entailments. Neither system is directly comparable to our own, since they employ different semantics, and in the case of the system of Du et al. target different DLs and queries.

The QuID system implements three approaches for query answering under IAR semantics: evaluation of the IAR perfect reformulation over the inconsistent ABox, evaluation over the ABox annotated with information about assertions that belong to some conflict of the original query enriched with conditions to filter out such assertions, and evaluation of the original query over the ABox from which every assertion involved in some conflict has been removed before query answering time. Our approach regarding IAR is closer to the two last, since our preprocessing phase where we compute and store the conflicts is similar in spirit to the annotation and extraction of the intersection of the repairs. The main difference is that we do not modify the query to retrieve only IAR-answers at evaluation time but rather filter out IAR-answers from candidate answers by checking a posteriori that they have causes without conflict.

We can observe some high-level similarities with Du et al.'s system which also has a preprocessing phase that compiles the KB, then employs SAT solvers and uses a reachability analysis to identify a query-relevant portion of the KB to do query answering.

There are also a few systems for querying inconsistent relational databases. Most relevant to our work is EQUIP [Kolaitis *et al.* 2013], which reduces AR conjunctive query answering in the presence of primary key constraints to binary integer programming (BIP). Similarly to our system, EQUIP first computes the IAR-answers and the causes of the answers with their conflicts. The encoding for AR consists in a first part that encodes the repairs, enforcing that exactly one tuple of each group of same-key tuple is selected, and a second part that ensures that the repair contains no cause. The main difference with CQAPri is that instead of building and solving one encoding for each answer, only one is built using variables to represent the different answers, so that setting them to 1 trivializes the equations related to the causes of this answer. The answers that do not hold under AR semantics are computed iteratively, by minimizing the sum of the answers variables, that are then set to 1 when found not AR, until the system becomes unsatisfiable. We considered using BIP but were not convinced by our preliminary experiments. Some systems focus on cases where consistent query answering is tractable, for restricted types of constraints or queries, using first-order rewritings (ConQuer[Fuxman & Miller 2005, Fuxman *et al.* 2005]) or conflict-hypergraph (Hippo [Chomicki *et al.* 2004a, Chomicki *et al.* 2004b]). Others handle more general constraints that can lead to different kinds of repairs, since for databases repairs it may be necessary to insert or modify tuples to restore consistency. For instance, ConsEx [Marileo & Bertossi 2010] reduces AR query answering to answer set programming (ASP) by building repair programs such that there is a one-to-one correspondence between stable models and repairs. Experiments reported in [Kolaitis *et al.* 2013] show that EQUIP outperforms ConsEx on its restricted setting that is closer to ours.

3.4.2 Experimental settings involving inconsistent DL-Lite KBs

The QuID benchmark QuID is evaluated using the LUBM TBox containing 43 concepts, 25 roles, 7 attributes and about 200 positive inclusions, approximated in DL-Lite by eliminating the inclusions that go beyond DL-Lite, then enriched with 10 negative inclusions, 5 identifications (that state that some sets of properties identify the instances of some basic concepts), and 3 denials constraints consisting in Boolean CQs (of the form $\exists\vec{y}\psi(\vec{y})$, where ψ is a conjunction of atoms using variables from $\vec{y}$) which have to be false.

Datasets are generated with the UBA Data Generator provided with LUBM, for 1, 5, 10 and 20 universities, leading to a size varying from about 100K to about 2 million assertions. For each of these consistent ABoxes, four inconsistent ABoxes are constructed by adding 1%, 5%, 10% and 20% of assertions that are in conflict. The main difference with our setting is the way conflicts are generated. Indeed, the original ABox is left consistent while each assertion added is in conflict with others. In practice, for a growing n, the same n fresh individuals are assigned to the eleven concepts that appear in some negative inclusion, and the same n pairs of fresh individuals are assigned to the five roles (or inverse roles) that appear in some negative inclusion. While such way of adding conflicts may make sense for evaluating the QuID system that implements IAR semantics, which only needs to ignore the assertions that are in some conflicts, it is not realistic at all because the new individuals are inserted in many concepts that are semantically very far from each other like Person, Publication, Course, and Organization. We could therefore not use these datasets to evaluate CQAPri since there is no chance that some query could hold under AR semantics but not under IAR semantics.

Du et al.'s benchmark In Du et al. later work on abduction over inconsistent DL-Lite KBs [Du *et al.* 2015], the Semintec and LUBM TBoxes without the non-DL-Lite axioms are used. The authors added negative inclusions to LUBM in a similar fashion to ours.

Regarding data, Semintec has a small ABox of about 65K assertions and the number of universities used for LUBM datasets ranges from 1 to 100 as in our experimental setting. Du et al. present in [Du *et al.* 2013] the tool called Injector they use to insert conflicts in ABoxes. Given a consistent KB and a number of conflicts to be inserted, Injector selects randomly a functional role or an atomic concept that has disjoint atomic concepts. If the KB already entails assertions that correspond to that role or concept, Injector selects randomly such an assertion and adds an assertion that conflicts it: in case of functional role, it relates the corresponding individual of the ABox to a new individual with that role, and in case of concept assertion, it assigns the individual to one of the disjoint atomic concepts. Otherwise, Injector adds two assertions that are in conflict in the same way using fresh individuals. This way of adding conflicts is much more realistic than that of QuID setting. Even if it is similar in spirit to ours, since it tries to distribute randomly conflicts over the ABox, there are some differences. First, we select randomly assertions of the initial ABox rather than concepts or roles that may be contradicted, which leads to a repartition of the conflicts that respects the structure of the data (since there may be lots of assertions of some concepts and no assertion of others). Second, we do not use only atomic concepts to build contradictions but also unqualified existential role restrictions. Finally we take into account another kind of possible errors by switching role individuals. In [Du *et al.* 2015], 0 to 400 "conflicts" (i.e. assertions that contradict an original assertion, or inconsistent pairs of assertions) are added to the consistent ABoxes. We prefer talk in terms of assertions involved in some conflicts rather than in terms of assertions added to the consistent ABox, since we can compute the ratio of conflicts of a real dataset, but not its ratio of erroneous facts. However, note that we added many more assertions than Du et al. in same size original datasets.

4

EXPLAINING INCONSISTENCY-TOLERANT QUERY ANSWERING

In this chapter, we address the problem of explaining why a tuple is or is not an answer to a query under the IAR, AR or brave semantics. We first define data-centric explanations for positive and negative answers. In the second section, we study their computational complexity in DL-Lite$_{\mathcal{R}}$, and propose algorithms to compute them by exploiting solvers for Boolean satisfaction and optimization problems. The third section presents our implementation within CQAPri and the experiments we conducted. Finally, the last section discusses the notion of responsibility in this context. The main results of this chapter have been published in [Bienvenu *et al.* 2016a].

4.1 Explaining query results

The need to equip reasoning systems with explanation services is widely acknowledged by the DL community (see Chapter 7 for more discussion and references), and such facilities are all the more essential when using inconsistency-tolerant semantics, as recently argued in [Arioua *et al.* 2014a, Arioua *et al.* 2014b, Arioua *et al.* 2015] which introduce an argumentation framework for explaining positive and negative answers under the ICR semantics. Indeed, the brave, AR, and IAR semantics allow one to classify query answers into three categories of increasing reliability (Possible, Likely and Sure), and a user may naturally wonder why a given tuple was assigned to, or excluded from, one of these categories. Our goal is therefore to help the user understand the classification of a particular tuple, e.g. why is $\vec{a}$ an AR-answer, and why is it not an IAR-answer? We address this issue by proposing and exploring a framework for explaining query answers under these three semantics by introducing the notion of *explanation* for *positive* and *negative query answers* under brave, AR, and IAR semantics.

Although the study of explanation services for DLs has thus far focused primarily on explaining entailed TBox axioms or ABox assertions, the problem of explaining conjunctive query answers under the classical semantics for consistent KBs has been studied. A proof-theoretic approach to explaining positive answers to CQs over DL-Lite$_{\mathcal{A}}$ KBs was introduced in [Borgida *et al.* 2008]. It outputs a single proof, involving both TBox axioms and ABox assertions, that is generated by "tracing back" the relevant part of the rewritten query, using minimality criteria to select a "simplest" proof. For negative answers, explanations for $\langle \mathcal{T}, \mathcal{A} \rangle \not\models q(\vec{a})$ are defined in [Calvanese *et al.* 2013] as sets $\mathcal{A}'$ of ABox assertions such that $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{A}' \rangle \models q(\vec{a})$. Practical algorithms and an implementation for computing such explanations were described in [Du *et al.* 2014]. The latter work was recently extended to the case of inconsistent KBs [Du *et al.* 2015]. Essentially the idea is to add a set of ABox assertions that will lead to the answer holding under IAR semantics (in particular, the new assertions must not introduce any inconsistencies).

To explain answers under inconsistency-tolerant semantics, the existing approaches are not sufficient. Indeed, it is no longer sufficient to prove that positive answers are entailed by some part of the ABox together with the TBox to show that they hold in every repair for instance. Regarding negative answers, they do not result from the absence of supporting facts anymore, but rather from the presence of conflicting assertions. Since we target scenarios in which inconsistencies are due to errors in the ABox, understanding the link between (possibly faulty) ABox assertions and query results is especially important. That is why we choose to focus on ABox assertions, rather than TBox axioms. The explanations we consider will therefore take either the form of a set of ABox assertions (viewed as a conjunction) or a set of sets of assertions (interpreted as a disjunction of conjunctions). We shall see that our "ABox-centric" explanation framework already poses non-trivial computational challenges. To get more complete explanations, which also present the TBox reasoning involved in the obtention of the results, our work could be combined with the framework of [Borgida *et al.* 2008].

Example 4.1.1. As a running example, consider the following inconsistent KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and queries.

$$\begin{aligned}
\mathcal{T} = &\{\mathsf{AProf} \sqsubseteq \mathsf{Prof}, \mathsf{FProf} \sqsubseteq \mathsf{Prof}, \exists \mathsf{Advise} \sqsubseteq \mathsf{Prof}, \mathsf{Prof} \sqsubseteq \mathsf{PhD}, \mathsf{Postdoc} \sqsubseteq \mathsf{PhD},\\
&\mathsf{AProf} \sqsubseteq \neg \mathsf{FProf}, \mathsf{Prof} \sqsubseteq \neg \mathsf{Postdoc}\}\\
\mathcal{A} = &\{\mathsf{Postdoc}(ann), \mathsf{AProf}(ann), \mathsf{FProf}(ann), \mathsf{Advise}(ann, bob),\\
&\mathsf{Teach}(ann, c_1), \mathsf{Teach}(ann, c_2), \mathsf{Teach}(ann, c_3)\}\\
q_1(x) = &\mathsf{Prof}(x)\\
q_2(x) = &\exists y \mathsf{PhD}(x) \wedge \mathsf{Teach}(x, y)\\
q_3(x) = &\exists y \mathsf{Teach}(x, y)
\end{aligned}$$

The repairs are as follows:

$$\begin{aligned}
&Rep(\mathcal{T}, \mathcal{A}) = \{\\
&\quad \{\mathsf{Postdoc}(ann), \mathsf{Teach}(ann, c_1), \mathsf{Teach}(ann, c_2), \mathsf{Teach}(ann, c_3)\},\\
&\quad \{\mathsf{AProf}(ann), \mathsf{Advise}(ann, bob), \mathsf{Teach}(ann, c_1), \mathsf{Teach}(ann, c_2), \mathsf{Teach}(ann, c_3)\},\\
&\quad \{\mathsf{FProf}(ann), \mathsf{Advise}(ann, bob), \mathsf{Teach}(ann, c_1), \mathsf{Teach}(ann, c_2), \mathsf{Teach}(ann, c_3)\}\\
&\}
\end{aligned}$$

Evaluating the queries over the KB yields the following results:

$\mathcal{K} \models_{\text{brave}} q_1(a)$ $\quad$ $\mathcal{K} \models_{\text{AR}} q_2(a)$ $\quad$ $\mathcal{K} \models_{\text{IAR}} q_3(a)$

$\mathcal{K} \not\models_{\text{AR}} q_1(a)$ $\quad$ $\mathcal{K} \not\models_{\text{IAR}} q_2(a)$ $\quad$ $\lhd$

The simplest answers to explain are positive brave- and IAR-answers. We can use the query's causes as explanations for the former, and the causes that do not participate in any contradiction for the latter. Indeed, we have seen in Chapter 2.2.2 that an answer holds under brave semantics just in the case that it has some cause, and under IAR semantics just in the case that it has some cause whose assertions do not belong to any conflict.

Definition 4.1.2 (Explanations for IAR- and brave-answers)**.** An explanation for $\mathcal{K} \models_{\text{brave}} q(\vec{a})$ is a cause for $q(\vec{a})$ w.r.t. $\mathcal{K}$. An explanation for $\mathcal{K} \models_{\text{IAR}} q(\vec{a})$ is a cause $\mathcal{C}$ for $q(\vec{a})$ w.r.t. $\mathcal{K}$ such that $\mathcal{C} \subseteq \mathcal{R}$ for every repair $\mathcal{R}$ of $\mathcal{K}$.

Example 4.1.3 (Example 4.1.1 cont'd)**.** There are three explanations for $\mathcal{K} \models_{\text{brave}} q_1(ann)$: $\mathsf{AProf}(ann)$, $\mathsf{FProf}(ann)$, and $\mathsf{Advise}(ann, bob)$.

There are twelve explanations for $\mathcal{K} \models_{\text{brave}} q_2(ann)$ that are the following conjunctions: $\mathsf{Postdoc}(ann) \wedge \mathsf{Teach}(ann, c_j)$, $\mathsf{AProf}(ann) \wedge \mathsf{Teach}(ann, c_j)$, $\mathsf{FProf}(ann) \wedge \mathsf{Teach}(ann, c_j)$, and $\mathsf{Advise}(ann, bob) \wedge \mathsf{Teach}(ann, c_j)$, for each $j \in \{1, 2, 3\}$.

There are three explanations for $\mathcal{K} \models_{\text{IAR}} q_3(ann)$: $\mathsf{Teach}(ann, c_1)$, $\mathsf{Teach}(ann, c_2)$, and $\mathsf{Teach}(ann, c_3)$, since these assertions are not involved in any conflict. $\lhd$

To explain why a tuple is an AR-answer, it is no longer sufficient to give a single cause, since the answer may be supported by different causes in different repairs. We will therefore define explanations as (minimal) disjunctions of causes that "cover" all repairs.

Definition 4.1.4 (Explanations for AR-answers)**.** An explanation for $\mathcal{K} \models_{\text{AR}} q(\vec{a})$ is a set $\mathcal{E} = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\} \subseteq \mathsf{causes}(q(\vec{a}), \mathcal{K})$ such that (i) every repair $\mathcal{R}$ of $\mathcal{K}$ contains some $\mathcal{C}_i$, and (ii) no proper subset of $\mathcal{E}$ satisfies this property.

Example 4.1.5 (Example 4.1.1 cont'd)**.** There are 36 explanations for $\mathcal{K} \models_{\text{AR}} q_2(ann)$, each taking one of the following two forms:

$$\begin{aligned}
\mathcal{E}_{ij} =& (\mathsf{Postdoc}(ann) \wedge \mathsf{Teach}(ann, c_i)) \vee (\mathsf{Advise}(ann, bob) \wedge \mathsf{Teach}(ann, c_j)) \\
\mathcal{E}'_{ijk} =& (\mathsf{Postdoc}(ann) \wedge \mathsf{Teach}(ann, c_i)) \vee (\mathsf{AProf}(ann) \wedge \mathsf{Teach}(ann, c_j)) \\
& \vee (\mathsf{FProf}(ann) \wedge \mathsf{Teach}(ann, c_k))
\end{aligned}$$

for some $i, j, k \in \{1, 2, 3\}$. $\lhd$

Remark 4.1.6 (Relationship with k-support semantics)**.** A tuple $\vec{a}$ is an answer to q under the k-support semantics if and only if $\mathcal{K} \models_{\text{AR}} q(\vec{a})$ has an explanation consisting of at most k causes.

We next consider how to explain negative AR- and IAR-answers, i.e., brave-answers not entailed under AR or IAR semantics. For AR semantics, the idea is to give a (minimal) subset of the ABox that is consistent with the TBox and contradicts every cause of the query, since any such subset can be extended to a repair that omits all causes. For IAR semantics, the formulation is slightly different as we only need to ensure that every cause is contradicted by some consistent subset, as this shows that no cause belongs to all repairs.

Definition 4.1.7 (Explanations for negative AR-answers)**.** An explanation for $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ is a $\mathcal{T}$-consistent subset $\mathcal{E} \subseteq \mathcal{A}$ such that: (i) $\langle \mathcal{T}, \mathcal{E} \cup \mathcal{C} \rangle \models \bot$ for every $\mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})$, (ii) no proper subset of $\mathcal{E}$ has this property.

Example 4.1.8 (Example 4.1.1 cont'd)**.** The unique explanation for $\mathcal{K} \not\models_{\text{AR}} q_1(ann)$ is $\mathsf{Postdoc}(ann)$, which contradicts the three causes of $q_1(ann)$. ◁

Remark 4.1.9 (Relationship with k-defeater semantics)**.** A tuple $\vec{a}$ is an answer to q under the k-defeater semantics if and only if there is no explanation of $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ of size at most k.

Definition 4.1.10 (Explanations for negative IAR-answers)**.** An explanation for $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ is a (possibly $\mathcal{T}$-inconsistent) subset $\mathcal{E} \subseteq \mathcal{A}$ such that: (i) for every $\mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})$, there exists a $\mathcal{T}$-consistent subset $\mathcal{E}' \subseteq \mathcal{E}$ with $\langle \mathcal{T}, \mathcal{E}' \cup \mathcal{C} \rangle \models \bot$, (ii) no proper subset of $\mathcal{E}$ has this property.

Example 4.1.11 (Example 4.1.1 cont'd)**.** The three explanations for $\mathcal{K} \not\models_{\text{IAR}} q_2(ann)$ are: $\mathsf{AProf}(ann) \wedge \mathsf{Postdoc}(ann)$, $\mathsf{FProf}(ann) \wedge \mathsf{Postdoc}(ann)$, and $\mathsf{Advise}(ann, bob) \wedge \mathsf{Postdoc}(ann)$, where the first assertion of each explanation contradicts the causes of $q_2(ann)$ that contain $\mathsf{Postdoc}(ann)$, and the second one contradicts those that contain $\mathsf{AProf}(ann)$, $\mathsf{FProf}(ann)$ or $\mathsf{Advise}(ann, bob)$. ◁

The following example illustrates that explanations are more informative than causes and conflicts, since some causes and conflicts may not be involved in the answer explanations. It also shows that explanations for negative answers should be accompanied with the explanations for being a brave-answer (i.e. the causes), because otherwise they may be difficult to understand.

Example 4.1.12. Consider the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

$$\begin{aligned}
\mathcal{T} = \{ & \exists \mathsf{Advise} \sqsubseteq \mathsf{Prof}, \mathsf{Prof} \sqsubseteq \mathsf{Employee}, \mathsf{Postdoc} \sqsubseteq \mathsf{Employee}, \mathsf{Prof} \sqsubseteq \neg \mathsf{Postdoc}, \\
& \exists \mathsf{WorkFor} \sqsubseteq \mathsf{Employee}, \exists \mathsf{WorkFor}^- \sqsubseteq \mathsf{Department}, \mathsf{Employee} \sqsubseteq \neg \mathsf{Department}, \\
& \exists \mathsf{TakeCourse}^- \sqsubseteq \mathsf{Course}, \exists \mathsf{Advise}^- \sqsubseteq \mathsf{Person}, \mathsf{Course} \sqsubseteq \neg \mathsf{Person} \} \\
\mathcal{A} = \{ & \mathsf{Postdoc}(ann), \mathsf{Advise}(ann, bob), \mathsf{Advise}(ann, carl), \mathsf{Teach}(ann, c_1), \\
& \mathsf{TakeCourse}(c_2, carl), \mathsf{WorkFor}(ann, dpt), \mathsf{WorkFor}(dpt, dan) \}
\end{aligned}$$

The conflicts are:

$$\begin{aligned}
\mathsf{conflicts}(\mathcal{K}) = \{ & \{\mathsf{Postdoc}(ann), \mathsf{Advise}(ann, bob)\}, \{\mathsf{Postdoc}(ann), \mathsf{Advise}(ann, carl)\}, \\
& \{\mathsf{Advise}(ann, carl), \mathsf{TakeCourse}(c_2, carl)\}, \\
& \{\mathsf{WorkFor}(ann, dpt), \mathsf{WorkFor}(dpt, dan)\} \}
\end{aligned}$$

Evaluating the query $q(x,y) = \mathsf{Employee}(x) \land \mathsf{Teach}(x,y)$ over $\mathcal{K}$ yields $\mathcal{K} \models_{\text{AR}} q(ann,c_1)$. The causes of $q(ann,c_1)$ are as follows:

$$\begin{aligned}&\mathsf{causes}(q(ann,c_1),\mathcal{K}) = \{\\&\quad\{\mathsf{Postdoc}(ann),\mathsf{Teach}(ann,c_1)\},\{\mathsf{Advise}(ann,bob),\mathsf{Teach}(ann,c_1)\},\\&\quad\{\mathsf{Advise}(ann,carl),\mathsf{Teach}(ann,c_1)\},\{\mathsf{WorkFor}(ann,dpt),\mathsf{Teach}(ann,c_1)\}\}\end{aligned}$$

There is only one explanation for $\mathcal{K} \models_{\text{AR}} q(ann,c_1)$: $(\mathsf{Postdoc}(ann) \land \mathsf{Teach}(ann,c_1)) \lor (\mathsf{Advise}(ann,bob) \land \mathsf{Teach}(ann,c_1))$. The cause $\mathsf{Advise}(ann,carl) \land \mathsf{Teach}(ann,c_1)$ does not participate in any explanation because it is conflicted by $\mathsf{TakeCourse}(c_2,carl)$ that can be chosen independently from the other causes. Indeed, note that if $\mathcal{R}$ is a repair of $\mathcal{K}$ that contains $\mathsf{Advise}(ann,carl) \land \mathsf{Teach}(ann,c_1)$, then $\mathcal{R}' = (\mathcal{R}\setminus\{\mathsf{Advise}(ann,carl)\}) \cup \{\mathsf{TakeCourse}(c_2,carl)\}$ is also a repair. Every explanation contains some cause $\mathcal{C}$ included in $\mathcal{R}'$, and $\mathsf{TakeCourse}(c_2,carl)$ does not belong to any cause, so $\mathcal{C} \subseteq \mathcal{R}$. It follows that $\mathsf{Advise}(ann,carl) \land \mathsf{Teach}(ann,c_1)$ is not needed to "cover" $\mathcal{R}$. For the same reason, the cause $\mathsf{WorkFor}(ann,dpt) \land \mathsf{Teach}(ann,c_1)$ does not belong to any explanation for $\mathcal{K} \models_{\text{AR}} q(ann,c_1)$ because of the assertion $\mathsf{WorkFor}(dpt,dan)$.

There are two explanations for $\mathcal{K} \not\models_{\text{IAR}} q(ann,c_1)$: $\mathsf{WorkFor}(dpt,dan) \land \mathsf{Postdoc}(ann) \land \mathsf{Advise}(ann,bob)$, and $\mathsf{WorkFor}(dpt,dan) \land \mathsf{Postdoc}(ann) \land \mathsf{Advise}(ann,carl)$. Note that $\mathsf{TakeCourse}(c_2,carl)$ is not involved in the explanations of $\mathcal{K} \not\models_{\text{IAR}} q(ann,c_1)$, while it is a conflict of the cause $\mathsf{Advise}(ann,carl) \land \mathsf{Teach}(ann,c_1)$. This is because $\mathsf{Postdoc}(ann)$ is also a conflict of this cause and is the only assertion that contradicts the other cause $\mathsf{Advise}(ann,bob) \land \mathsf{Teach}(ann,c_1)$, so each set of assertions which contradicts every cause contains $\mathsf{Postdoc}(ann)$, and $\mathsf{TakeCourse}(c_2,carl)$ is not needed, so does not appear in any minimal such set.

If a user receives the explanations for $\mathcal{K} \not\models_{\text{IAR}} q(ann,c_1)$, without the causes of $q(ann,c_1)$, it can be very hard for him to figure out why $\mathsf{WorkFor}(dpt,dan)$ is relevant. Indeed, there is no obvious relation between this assertion and the answer (ann,c_1), since $\mathsf{WorkFor}(dpt,dan)$ does not involve any of the individuals of the answer. Even if the explanations for $\mathcal{K} \models_{\text{AR}} q(ann,c_1)$ are provided, it would not help because neither dpt nor dan appears in them. To understand why $\mathsf{WorkFor}(dpt,dan)$ is part of the explanations for $\mathcal{K} \not\models_{\text{IAR}} q(ann,c_1)$, the user needs to know the cause $\{\mathsf{WorkFor}(ann,dpt),\mathsf{Teach}(ann,c_1)\}$. Ideally, we should allow the user to ask for a justification of the relevance of an assertion α of the explanations, and provide the causes (or at least one cause) that contain some assertion in conflict with α. $\triangleleft$

When there is a large number of explanations for a given answer, it may be impractical to present them all to the user. In such cases, instead of presenting all explanations, one may choose to rank the explanations according to some preference criteria, and to present one or a small number of most *preferred explanations*. In this work, we will use *cardinality* to rank explanations for brave- and IAR-answers and negative AR- and IAR-answers. For positive AR-answers, we consider two ranking criteria: the *number of disjuncts*, and the total *number of assertions*. Another interesting criterion would be the difficulty of the associated TBox reasoning. For example, we may compute for each cause the minimum number of TBox axioms needed to show that the cause yields the query, and then use this number to rank explanations for brave- and IAR-answers.

Example 4.1.13 (Example 4.1.1 cont'd)**.** Reconsider explanations $\mathcal{E}_{11}$ and $\mathcal{E}'_{123}$ for $\mathcal{K} \models_{\text{AR}} q_2(ann)$. There are at least two reasons why $\mathcal{E}_{11}$ may be considered easier to understand than $\mathcal{E}'_{123}$. First, $\mathcal{E}_{11}$ contains fewer disjuncts, hence requires less disjunctive reasoning. Second, both disjuncts of $\mathcal{E}_{11}$ use the same Teach assertion, whereas $\mathcal{E}'_{123}$ uses three different Teach assertions, which may lead the user to (wrongly) believe all are needed to obtain the query result. Preferring explanations having the fewest number of disjuncts, and among them, those involving a minimal set of assertions, leads to focusing on the explanations of the form $\mathcal{E}_{ii}$, where $i \in \{1,2,3\}$. ◁

A second complementary approach to dealing with a large number of explanations is to concisely summarize the set of explanations in terms of the *necessary assertions* (i.e. appearing in every explanation) and the *relevant assertions* (i.e. appearing in at least one explanation). The advantage of this approach is to present the whole information to the user without overwhelming him with all explanations. This is especially relevant in the case of positive AR and negative answers, where the number of explanations may be exponential in the size of the ABox because of the combination of the different causes for AR-answers, and ways of contradicting each cause for negative answers. Indeed, the set of conflicts of each cause can be as large as the ABox. Even for positive IAR or brave-answers, the number of explanations can be very large (imagine for instance a query that retrieves the persons who teach something, advise some students and have some publications).

Example 4.1.14 (Example 4.1.1 cont'd)**.** If we tweak the example KB to include n courses taught by ann, then there would be $n^2 + n^3$ explanations of the form $\mathcal{E}_{ij}$ and $\mathcal{E}'_{ijk}$ for $\mathcal{K} \models_{\text{AR}} q_2(ann)$, built using only $n+4$ assertions. Presenting the necessary assertions (here: $\mathsf{Postdoc}(ann)$) and relevant ones ($\mathsf{AProf}(ann)$, $\mathsf{FProf}(ann)$, $\mathsf{Advise}(ann, bob)$, and $\mathsf{Teach}(ann, c_j)$ for $1 \leq j \leq n$) gives a succinct overview of the set of explanations. ◁

4.2 Complexity analysis and algorithms

In this section, we study the computational properties of the different notions of explanation for DL-Lite$_{\mathcal{R}}$ KBs. In addition to the problem of generating a single explanation (GENONE), or a single best explanation (GENBEST) according to a given criteria, we consider four related decision problems: decide whether a given assertion appears in some explanation (REL) or in every explanation (NEC), decide whether a candidate is an explanation (REC), resp. a best explanation according a given criterion (BEST REC).

This section aims at proving the following theorem and provides the algorithms we use to explain answers.

Theorem 4.2.1. *The complexity results displayed in Table 4.1 hold.*

When showing that a decision problem is hard for a given complexity class, we use standard polynomial-time many-one reductions (also known as Karp reductions), which transform an instance of one decision problem into an instance of a second decision problem.

Table 4.1 Data complexity results for CQs explanations in DL-Lite$_{\mathcal{R}}$.

	brave, IAR	**AR**	**neg. AR**	**neg. IAR**
GENONE	in P	NP-h	NP-h	in P
GENBEST†	in P	Σ_2^p-h‡	NP-h	NP-h*
REL	in P	Σ_2^p-co	NP-co	in P
NEC	in P	NP-co	coNP-co	in P
REC	in P	BH_2-co	in P	in P
BEST REC†	in P	Π_2^p-co‡	coNP-co*	coNP-co*

† upper bounds hold for ranking criteria that can be decided in P
‡ lower bounds hold for smallest disjunction or fewest assertions
* lower bounds hold for cardinality-minimal explanations

We consider that a procedure solves the generation task GENONE (resp. GENBEST) if it outputs an explanation (resp. best explanation according to the chosen criterion) when there is at least one explanation, and otherwise, it outputs no. To show that a generation task is hard for a class $\mathcal{C}$, we reduce a $\mathcal{C}$-hard decision problem to it. As we cannot use many-one reductions (which relate two decision problems), we will use polynomial-time Turing reductions, that is, we will show how to solve the $\mathcal{C}$-hard decision problem using a polynomial-time Turing machine that can use the generation task as an oracle. Moreover, to prove a stronger intractability result, we will only allow a single oracle call.

4.2.1 Positive brave and IAR-answers

We recall that in DL-Lite$_{\mathcal{R}}$, the conflicts for a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and the causes of a query q can be computed in polynomial time w.r.t. data complexity. Since the explanations for positive brave-answers are the causes, and the explanations for positive IAR-answers are the causes that belong to every repair, i.e. those which do not contain any assertion involved in a conflict of $\mathcal{K}$, it is possible to compute the entire set of explanations for positive brave and IAR-answers in P. This means that GENONE is in P. For polynomial-time ranking criteria, both GENBEST and BEST REC are solvable in P since we can compare all of the explanations to identify the best ones. The sets of relevant and necessary assertions can be computed in P by taking the union and intersection of the explanations.

4.2.2 Positive AR-answers

We relate explanations of positive AR-answers to minimal unsatisfiable subsets of a set of propositional clauses.

Definition 4.2.2 (Minimal Unsatisfiable Subset)**.** Given sets F and H of soft and hard clauses respectively, a subset $M \subseteq F$ is a *minimal unsatisfiable subset* (MUS) of F w.r.t. H if (i) $M \cup H$ is unsatisfiable, and (ii) $M' \cup H$ is satisfiable for every $M' \subsetneq M$.

To explain $\mathcal{K} \models_{\text{AR}} q(\vec{a})$, we exploit the encoding of Figure 3.1. We consider the soft clauses

$$\varphi_{\neg q} = \{\lambda_{\mathcal{C}} \mid \mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})\} \text{ with } \lambda_{\mathcal{C}} = \bigvee_{\beta \in \mathsf{confl}(\mathcal{C}, \mathcal{K})} x_\beta$$

that try to contradict every cause, and the hard clauses

$$\varphi_{cons} = \{\neg x_\alpha \vee \neg x_\beta \mid x_\alpha, x_\beta \in \mathsf{vars}(\varphi_{\neg q}), \beta \in \mathsf{confl}(\{\alpha\}, \mathcal{K})\}$$

that enforce consistency.

Proposition 4.2.3. *A set $\mathcal{E} \subseteq \mathsf{causes}(q(\vec{a}), \mathcal{K})$ is an explanation for $\mathcal{K} \models_{AR} q(\vec{a})$ iff $\{\lambda_{\mathcal{C}} \mid \mathcal{C} \in \mathcal{E}\}$ is a MUS of $\varphi_{\neg q}$ w.r.t. φ_{cons}.*

Proof. First suppose that $\mathcal{E}$ is an explanation of $\mathcal{K} \models_{\text{AR}} q(\vec{a})$. By Definition 4.1.4, this means that every repair of $\mathcal{K}$ contains at least one cause $\mathcal{C}$ from $\mathcal{E}$. It follows that it is not possible to select one conflicting assertion for each corresponding cause in a consistent way, i.e. $\{\lambda_{\mathcal{C}} \mid \mathcal{C} \in \mathcal{E}\} \cup \varphi_{cons}$ is inconsistent. Moreover, the minimality condition ensures that for every proper subset $\mathcal{E}' \subsetneq \mathcal{E}$, there is a repair $\mathcal{R}$ that does not contain any cause from $\mathcal{E}'$. We can use $\mathcal{R}$ to select a consistent set of assertions that conflict with every cause in $\mathcal{E}'$, which means that $\{\lambda_{\mathcal{C}} \mid \mathcal{C} \in \mathcal{E}'\} \cup \varphi_{cons}$ is satisfiable. Thus, $\{\lambda_{\mathcal{C}} \mid \mathcal{C} \in \mathcal{E}\}$ is a MUS of $\varphi_{\neg q}$ w.r.t. φ_{cons}.

Conversely, suppose that $\{\lambda_{\mathcal{C}} \mid \mathcal{C} \in \mathcal{E}\}$ is a MUS of $\varphi_{\neg q}$ w.r.t. φ_{cons}. Then $\{\lambda_{\mathcal{C}} \mid \mathcal{C} \in \mathcal{E}\} \cup \varphi_{cons}$ is unsatisfiable, and every $\{\lambda_{\mathcal{C}} \mid \mathcal{C} \in \mathcal{E}'\} \cup \varphi_{cons}$ with $\mathcal{E}' \subsetneq \mathcal{E}$ is satisfiable. The fact that $\{\lambda_{\mathcal{C}} \mid \mathcal{C} \in \mathcal{E}\} \cup \varphi_{cons}$ is unsatisfiable means that there is no way to consistently contradict the causes in $\mathcal{E}$, so every repair must contain one of the causes in $\mathcal{E}$. The satisfiability of $\{\lambda_{\mathcal{C}} \mid \mathcal{C} \in \mathcal{E}'\} \cup \varphi_{cons}$ for $\mathcal{E}' \subsetneq \mathcal{E}$ implies the existence of a repair that omits every cause in $\mathcal{E}'$. We have thus shown that $\mathcal{E}$ satisfies the conditions of Definition 4.1.4, so it is an explanation of $\mathcal{K} \models_{\text{AR}} q(\vec{a})$. □

Using Proposition 4.2.3 and known complexity results for MUSes yields the following upper bounds:

Proposition 4.2.4. *Regarding explanations for AR-answers,* REC *is in* BH_2*,* BEST REC *is in* Π_2^p*, and* REL *is in* Σ_2^p *w.r.t. data complexity.*

Proof. We recall the following complexity results for MUSes (see [Liberatore 2005]):

- Deciding if a set of clauses is a MUS is BH_2-complete.
- Deciding if a clause belongs to some MUS is Σ_2^p-complete.

When combined with Proposition 4.2.3, the first item yields membership in BH_2 of REC. For BEST REC, we show that an explanation is *not* a best one by guessing a better candidate and checking in BH_2 that it is an explanation. This yields a Σ_2^p procedure for the complement of BEST REC, hence membership in Π_2^p for BEST REC.

For REL, we note that an assertion α is relevant for explaining $\mathcal{K} \models_{\text{AR}} q(\vec{a})$ just in the case that there exists a cause $\mathcal{C}$ for $q(\vec{a})$ w.r.t. $\mathcal{K}$ that contains α and appears in some explanation. By Proposition 4.2.3, the latter holds just in the case that $\lambda_{\mathcal{C}}$ belongs to some MUS of $\varphi_{\neg q}$ w.r.t. φ_{cons}. By the second item above, deciding whether a particular clause $\lambda_{\mathcal{C}}$ belongs to some MUS can be decided in Σ_2^p. To obtain a Σ_2^p decision procedure for REL, we simply add an initial non-deterministic guess of a cause $\mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})$ that mentions the considered assertion α. □

We next show the NP upper bound for NEC.

Proposition 4.2.5. *Regarding explanations for positive AR-answers,* NEC *is in* NP *w.r.t. data complexity.*

Proof. If α is belongs to every explanation of $\mathcal{K} \models_{\text{AR}} q(\vec{a})$, then there exists a repair $\mathcal{R}$ such that every cause for $q(\vec{a})$ included in $\mathcal{R}$ contains α. Otherwise, if for every repair $\mathcal{R}$, there was a cause $\mathcal{C}_{\mathcal{R}}$ such that $\alpha \notin \mathcal{C}_{\mathcal{R}}$, then there would be a minimal disjunction of these $\mathcal{C}_{\mathcal{R}}$ which covers every repair and does not contain α. In the other direction, it is clear that if there exists such a repair $\mathcal{R}$, then any minimal disjunction of causes that covers every repair contains α.

It follows that α belongs to every explanation of $\mathcal{K} \models_{\text{AR}} q(\vec{a})$ just in the case that either there are no explanations at all (i.e. $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$) or there exists a repair $\mathcal{R}$ of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ such that $\langle \mathcal{T}, \mathcal{R} \setminus \{\alpha\} \rangle \not\models q(\vec{a})$.

Both conditions can be tested in NP w.r.t. data complexity. Indeed, to decide whether the second condition holds, we simply guess a subset $\mathcal{R} \subseteq \mathcal{A}$ and check (in P w.r.t. data complexity) that $\mathcal{R}$ is a repair and $\langle \mathcal{T}, \mathcal{R} \setminus \{\alpha\} \rangle \not\models q(\vec{a})$. □

The following proposition shows how the connection to MUSes can be exploited to obtain matching lower bounds.

Proposition 4.2.6. *Regarding explanations for positive AR-answers,* REC *is* BH_2*-hard,* NEC *is* NP*-hard,* REL *is* Σ_2^p*-hard, and* GENONE *is* NP*-hard w.r.t. data complexity. Moreover, if we rank explanations according to the number of causes or number of assertions, then* BEST REC *(resp.* GENBEST*) is* Π_2^p*-hard (resp.* Σ_2^p*-hard) w.r.t. data complexity. .*

Proof. We show how the MUSes of a propositional clause set can be captured by explanations of positive AR-answers.

Let $\varphi_0 = \{C_1, ..., C_k\}$ be a set of clauses over $\{X_1, ..., X_n\}$. Consider the KB and query used in the reduction of the proof of coNP-hardness of AR entailment (Theorem 2.2.6):

$$\begin{aligned}
\mathcal{T}_0 &= \{\exists P^- \sqsubseteq \neg \exists N^-, \exists P \sqsubseteq \neg \exists U^-, \exists N \sqsubseteq \neg \exists U^-, \exists U \sqsubseteq A\} \\
\mathcal{A}_0 &= \{P(c_j, x_i) \mid X_i \in C_j\} \cup \{N(c_j, x_i) \mid \neg X_i \in C_j\} \cup \{U(a, c_j) \mid 1 \leq j \leq k\} \\
q_0 &= A(x)
\end{aligned}$$

The causes for $q_0(a)$ are given by the assertions $U(a, c_j)$, which are in conflict with assertions of the form $P(c_j, x_i)$ or $N(c_j, x_i)$. It was shown that $\langle \mathcal{T}_0, \mathcal{A}_0 \rangle \models_{\mathsf{AR}} A(a)$ iff φ_0 is unsatisfiable. To prove the proposition, we will require the following stronger claim:

Claim. The following are equivalent:

1. the set of clauses $\{C_{j_1}, ..., C_{j_p}\}$ is unsatisfiable

2. every repair of $\langle \mathcal{T}_0, \mathcal{A}_0 \rangle$ contains some assertion from $\{U(a, c_{j_1}), ..., U(a, c_{j_p})\}$

Proof of claim. It will be more convenient to show that the negations of the two statements are equivalent. First suppose that $\{C_{j_1}, ..., C_{j_p}\}$ is satisfiable, as witnessed by the satisfying assignment ν. Define a repair $\mathcal{R}_\nu$ of $\langle \mathcal{T}_0, \mathcal{A}_0 \rangle$ by including the assertion $P(c_j, v_i)$ if $\nu(v_i) =$ true, including $N(c_j, v_i)$ if $\nu(v_i) = \mathsf{false}$, and then adding as many other assertions as needed to obtain a maximal $\mathcal{T}_0$-consistent subset. Since ν satisfies every clause in $\{C_{j_1}, ..., C_{j_p}\}$, it follows that for every index $\ell \in \{j_1, \ldots, j_p\}$, the clause C_ℓ contains a positive literal v_ℓ such that $\nu(v_\ell) = \mathsf{true}$, or a negative literal $\neg v_\ell$ such that $\nu(v_\ell) = \mathsf{false}$. In the former case, $\mathcal{R}_\nu$ contains the assertion $P(c_\ell, v_\ell)$, and in the latter case, $\mathcal{R}_\nu$ contains $N(c_\ell, v_\ell)$. In both cases, there is an assertion in $\mathcal{R}_\nu$ that conflicts with $U(a, c_\ell)$, so the latter assertion cannot appear in $\mathcal{R}_\nu$. We have thus shown that $\mathcal{R}_\nu$ does not contain any of the assertions in $\{U(a, c_{j_1}), ..., U(a, c_{j_p})\}$.

Next suppose there exists a repair $\mathcal{R}$ of $\langle \mathcal{T}_0, \mathcal{A}_0 \rangle$ that has an empty intersection with the set $\{U(a, c_{j_1}), ..., U(a, c_{j_p})\}$. By the maximality of $\mathcal{R}$, it follows that for every $\ell \in \{j_1, \ldots, j_p\}$, there must exist an assertion in $\mathcal{R}$ of the form $P(c_\ell, v_i)$ or $N(c_\ell, v_i)$. Define a (possibly partial) assignment $\nu_\mathcal{R}$ by setting by X_i to true if $\mathcal{R}$ contains some $P(c_j, x_i)$ and to false if $\mathcal{R}$ contains some $N(c_j, x_i)$ (recall that $\mathcal{R}$ is consistent with $\mathcal{T}_0$, and so it cannot contain both $P(c_j, x_i)$ and $N(c_{j'}, x_i)$). By construction, $\nu_\mathcal{R}$ satisfies all of the clauses in $\{C_{j_1}, ..., C_{j_p}\}$, i.e. $\{C_{j_1}, ..., C_{j_p}\}$ is satisfiable. (*end proof of claim*)

It follows from the preceding claim that the explanations for $\langle \mathcal{T}_0, \mathcal{A}_0 \rangle \models_{\mathsf{AR}} q_0(a)$, i.e. the minimal sets of causes for $q_0(a)$ that cover all repairs, correspond precisely to the MUSes of φ_0. We can therefore exploit known complexity results for MUSes [Liberatore 2005]:

- Deciding if a clause belongs to a MUS is Σ^p_2-complete, so deciding if $U(a, c_j)$ belongs to an explanation for $\langle \mathcal{T}_0, \mathcal{A}_0 \rangle \models_{\mathsf{AR}} q_0(a)$ is Σ^p_2-hard w.r.t. data complexity. Thus, we have a Σ^p_2 lower bound for REL.
- Deciding if a clause belongs to every MUS is NP-complete, so deciding if $U(a, c_j)$ belongs to every explanation for $\langle \mathcal{T}_0, \mathcal{A}_0 \rangle \models_{\mathsf{AR}} q_0(a)$ is NP-hard w.r.t. data complexity. This gives an NP lower bound for NEC.
- REC: Deciding if a set of clauses is a MUS is BH_2-complete, so deciding if $\{\{U(a, c_{j_1})\}, ..., \{U(a, c_{j_p})\}\}$ is an explanation is BH_2-hard w.r.t. data complexity. Hence, REC is BH_2-hard.

The proof of [Liberatore 2005] for Σ^p_2-hardness of deciding if there exists a MUS of size at most k also shows that deciding if a set of clauses is a smallest MUS is Π^p_2-hard. It follows that deciding if an explanation for an AR-answer contains a smallest number of causes is Π^p_2-hard. Moreover, since every cause in the considered KB consists of a single assertion, deciding if an explanation for an AR-answer contains a smallest number of assertions is also Π_2^p-hard.

To see why the generation task GENONE is NP-hard, we note that to solve the NP-complete problem of whether $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$, it suffices to call the procedure for GENONE to generate a single explanation for $\mathcal{K} \models_{\text{AR}} q(\vec{a})$. If the procedure outputs 'no' (meaning there is no explanation for $\mathcal{K} \models_{\text{AR}} q(\vec{a})$), then we output 'yes', and if it outputs an explanation, then we return 'no'.

The Σ_2^p-hardness of GENBEST, when explanations are ranked based upon the number of disjuncts or the number of assertions, follows from the Π_2^p-hardness of BEST REC for these same criteria. Indeed, to show that an explanation is *not* a best explanation, it suffices to generate a best explanation (GENBEST) and verify that it has fewer disjuncts / assertions than the explanation at hand. □

4.2.3 Negative AR-answers

We relate explanations of negative AR-answers to minimal models of $\varphi_{\neg q} \cup \varphi_{cons}$.

Definition 4.2.7 (Minimal model)**.** Given a clause set ψ over variables X, a set $M \subseteq X$ is a *minimal model* of ψ iff (i) every valuation that assigns true to all variables in M satisfies ψ, (ii) no $M' \subsetneq M$ satisfies this condition. Cardinality-minimal models are defined analogously.

Proposition 4.2.8. *A set $\mathcal{E}$ is an explanation (resp. cardinality-minimal explanation) for $\mathcal{K} \not\models_{AR} q(\vec{a})$ iff $\{x_\alpha \mid \alpha \in \mathcal{E}\}$ is a minimal (resp. cardinality-minimal) model of $\varphi_{\neg q} \cup \varphi_{cons}$.*

Proof. We recall that the reason why $\mathcal{K} \models_{\text{AR}} q(\vec{a})$ iff $\varphi_{\neg q} \wedge \varphi_{cons}$ is unsatisfiable is because the assertions whose corresponding variables are assigned to true in a valuation that satisfies $\varphi_{\neg q} \wedge \varphi_{cons}$ form a subset of the ABox which: (i) contradicts every cause, since $\varphi_{\neg q}$ states that for every cause, one conflicting assertion is selected, and (ii) is consistent, since φ_{cons} states that two assertions in a conflict cannot be selected together. Thus, the inclusion-minimal models of $\varphi_{\neg q} \wedge \varphi_{cons}$ are precisely the explanations for negative AR-answers. □

Next we show the complexity upper bounds for the decision problems.

Proposition 4.2.9. *Regarding explanations for negative AR-answers,* REC *is in* P*,* BEST REC *is in co*NP*,* REL *is in* NP*, and* NEC *is in co*NP *w.r.t. data complexity.*

Proof. It follows from Definition 4.1.7 that deciding whether $\mathcal{E} \subseteq \mathcal{A}$ is an explanation for $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ can be done in P (data complexity) by checking:

- consistency of $\langle \mathcal{T}, \mathcal{E} \rangle$
- inconsistency of $\langle \mathcal{T}, \mathcal{E} \cup \mathcal{C} \rangle$ for every $\mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})$
- minimality of $\mathcal{E}$: no proper subset $\mathcal{E}' \subsetneq \mathcal{E}$ satisfies the two previous conditions.

We can decide in NP that an explanation $\mathcal{E}$ is *not* a best explanation (according to some polynomial-time ranking criterion) by guessing a subset $\mathcal{E}' \subseteq \mathcal{A}$ and verifying in P w.r.t. data complexity that $\mathcal{E}'$ is an explanation (see previous paragraph) and that it is better than $\mathcal{E}$ according to the given criterion. This yields a coNP upper bound for BEST REC.

A simple NP procedure for deciding REL (resp. *not* NEC) consists in guessing a subset $\mathcal{E} \subseteq \mathcal{A}$ that contains (does not contain) the considered assertion and checking in P whether it is an explanation (using the P procedure for REC). □

For the purposes of implementation, we propose alternative procedures for REL and NEC.

Proposition 4.2.10. *An assertion* α *is relevant for explaining* $\mathcal{K} \not\models_{AR} q(\vec{a})$ *iff the clause set* $\varphi_{\neg q} \cup \varphi_{cons} \cup \varphi_{\alpha}$ *is satisfiable, where*

$$\varphi_\alpha = \{ \bigvee_{\mathcal{C} \in \mathsf{causes}(q(\vec{a}),\mathcal{K}), \alpha \in \mathsf{confl}(\mathcal{C},\mathcal{K})} x_{\mathcal{C}} \} \cup$$
$$\{\neg x_{\mathcal{C}} \vee \neg x_\beta \mid \mathcal{C} \in \mathsf{causes}(q(\vec{a}),\mathcal{K}), \alpha \in \mathsf{confl}(\mathcal{C},\mathcal{K}), \beta \in \mathsf{confl}(\mathcal{C},\mathcal{K}), \beta \neq \alpha\}$$

Proof. If α is relevant, there exists an explanation $\mathcal{E}$ such that $\alpha \in \mathcal{E}$. Since $\mathcal{E}$ is minimal, there exists a cause $\mathcal{C}$ such that $\mathcal{C} \cup (\mathcal{E} \setminus \{\alpha\})$ is consistent. It follows that no assertion $\beta \in \mathsf{confl}(\mathcal{C},\mathcal{K})$ belongs to $\mathcal{E}$ except for α. Then the valuation ν such that $\nu(x_{\mathcal{C}}) = \mathsf{true}$, and for every assertion β, $\nu(x_\beta) = \mathsf{true}$ if $\beta \in \mathcal{E}$, $\nu(x_\beta) = \mathsf{false}$ otherwise, satisfies $\varphi_{\neg q} \cup \varphi_{cons} \cup \varphi_\alpha$.

In the other direction, if $\varphi_{\neg q} \cup \varphi_{cons} \cup \varphi_\alpha$ is satisfiable, it is possible to contradict every cause with a consistent set $\mathcal{E}$ of assertions such that there exists a cause $\mathcal{C}$ such that the only assertion of $\mathcal{E} \cap \mathsf{confl}(\mathcal{C},\mathcal{K})$ is α. Then an explanation that contains α is included in $\mathcal{E}$. □

Proposition 4.2.11. *An assertion* α *is necessary for explaining* $\mathcal{K} \not\models_{AR} q(\vec{a})$ *iff the set of clauses* $\varphi_{\neg q} \cup \varphi_{cons} \cup \{\neg x_\alpha\}$ *is unsatisfiable.*

Proof. By Proposition 4.2.8, $\mathcal{E}$ is an explanation for $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ iff $\{x_\alpha \mid \alpha \in \mathcal{E}\}$ is a minimal model of $\varphi_{\neg q} \cup \varphi_{cons}$. It follows that α belongs to every explanation for $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ just in the case that x_α belongs to every minimal model of $\varphi_{\neg q} \cup \varphi_{cons}$, so $\varphi_{\neg q} \cup \varphi_{cons} \cup \{\neg x_\alpha\}$ has no model, i.e. is unsatisfiable. □

The next proposition establishes matching lower bounds. All reductions are illustrated on Figure 4.1.

Proposition 4.2.12. *Regarding explanations for negative AR-answers,* NEC *is co*NP*-hard, and* REL*,* GENONE*, and* GENBEST *(for any ranking criterion) are* NP*-hard w.r.t. data complexity. If explanations are ranked by cardinality, then* BEST REC *is co*NP*-hard w.r.t. data complexity.*

Proof. All reductions are from (UN)SAT. Let $\varphi = C_1 \wedge ... \wedge C_k$ be a set of clauses over propositional variables $\{X_1, ..., X_n\}$.

• GENONE and GENBEST: Let $\mathcal{T}_0$, $\mathcal{A}_0$, and q_0 be as in Proposition 4.2.6. We know that φ_0 is satisfiable iff $\langle \mathcal{T}_0, \mathcal{A}_0 \rangle \not\models_{\text{AR}} A(a)$. Thus, to decide the satisfiability of φ_0, we generate a (best) explanation of $\langle \mathcal{T}_0, \mathcal{A}_0 \rangle \not\models_{\text{AR}} A(a)$. If an explanation is produced, then we return 'yes', and if the procedure returns with no explanation, then we output 'no'.

• NEC: We again let $\mathcal{T}_0$, $\mathcal{A}_0$, and q_0 be as in Proposition 4.2.6. Define a new TBox $\mathcal{T}_1 = \mathcal{T}_0 \cup \{\exists U \sqsubseteq \neg S\}$ and ABox $\mathcal{A}_1 = \mathcal{A}_0 \cup \{S(a)\}$. By construction, the assertion $S(a)$ contradicts every cause for $q_0(a)$, so $\langle \mathcal{T}_1, \mathcal{A}_1 \rangle \not\models_{\text{AR}} q_0(a)$. We show that deciding whether φ is satisfiable is equivalent to deciding if $S(a)$ is *not* necessary for explaining $\langle \mathcal{T}_1, \mathcal{A}_1 \rangle \not\models_{\text{AR}} q_0(a)$. This establishes the coNP-hardness of checking necessity.

$\Rightarrow$ Let ν be a satisfying valuation for φ. It can be easily verified that the set $\{P(c_j, v_i) \in \mathcal{A}_0 \mid \nu(v_i) = \mathsf{true}\} \cup \{N(c_j, v_i) \in \mathcal{A}_0 \mid \nu(v_i) = \mathsf{false}\}$ conflicts with every cause of $q_0(a)$, and so by choosing a subset of these assertions, we can construct an explanation for $\langle \mathcal{T}_1, \mathcal{A}_1 \rangle \not\models_{\text{AR}} q_0(a)$ that does not contain $S(a)$.

$\Leftarrow$ An explanation $\mathcal{E}$ that does not contain $S(a)$ forms a $\mathcal{T}_1$-consistent set of P- and N-assertions such that every c_j has an outgoing P- or N-edge. We obtain a (partial) assignment $\nu_{\mathcal{E}}$ that satisfies φ by setting $\nu_{\mathcal{E}}(v_i) = \mathsf{true}$ if $\mathcal{E}$ contains an assertion $P(c_j, v_i)$ and $\nu_{\mathcal{E}}(v_i) = \mathsf{false}$ if $\mathcal{E}$ contains an assertion $N(c_j, v_i)$.

• REL: We use the TBox $\mathcal{T}_1$ and the ABox $\mathcal{A}_2 = \mathcal{A}_1 \cup \{U(a, c_{k+1}), P(c_{k+1}, x_{n+1})\}$. Again, we note that $S(a)$ contradicts every cause for $q_0(a)$, so $\langle \mathcal{T}_1, \mathcal{A}_2 \rangle \not\models_{\text{AR}} q_0(a)$. We show that φ is satisfiable iff $P(c_{k+1}, x_{n+1})$ is relevant for explaining $\langle \mathcal{T}_1, \mathcal{A}_2 \rangle \not\models_{\text{AR}} q_0(a)$; it follows that relevance is NP-hard.

$\Rightarrow$ If φ is satisfiable, then we can obtain an explanation for $\langle \mathcal{T}_1, \mathcal{A}_2 \rangle \not\models_{\text{AR}} q_0(a)$ by adding $P(c_{k+1}, x_{n+1})$ to a minimal subset of the P- and N-assertions corresponding to a satisfying truth assignment for φ.

$\Leftarrow$ If φ is unsatisfiable, then every explanation must contain $S(a)$. It follows that $\{S(a)\}$ is the only explanation, so $P(c_{k+1}, x_{n+1})$ is not relevant.

• BEST REC: We consider the following KB:

$$\begin{aligned}
\mathcal{T}_3 &= \mathcal{T}_0 \cup \{U_1 \sqsubseteq U, U_2 \sqsubseteq U, \exists U_1^- \sqsubseteq \neg T, \exists U_2 \sqsubseteq \neg S\} \\
\mathcal{A}_3 &= \{P(c_j, x_i) \mid X_i \in C_j\} \cup \{N(c_j, x_i) \mid \neg X_i \in C_j\} \cup \\
&\quad \{U_1(a, c_j), U_2(a, c_j), T(c_j) \mid 1 \leq j \leq k\} \cup \{S(a)\}
\end{aligned}$$

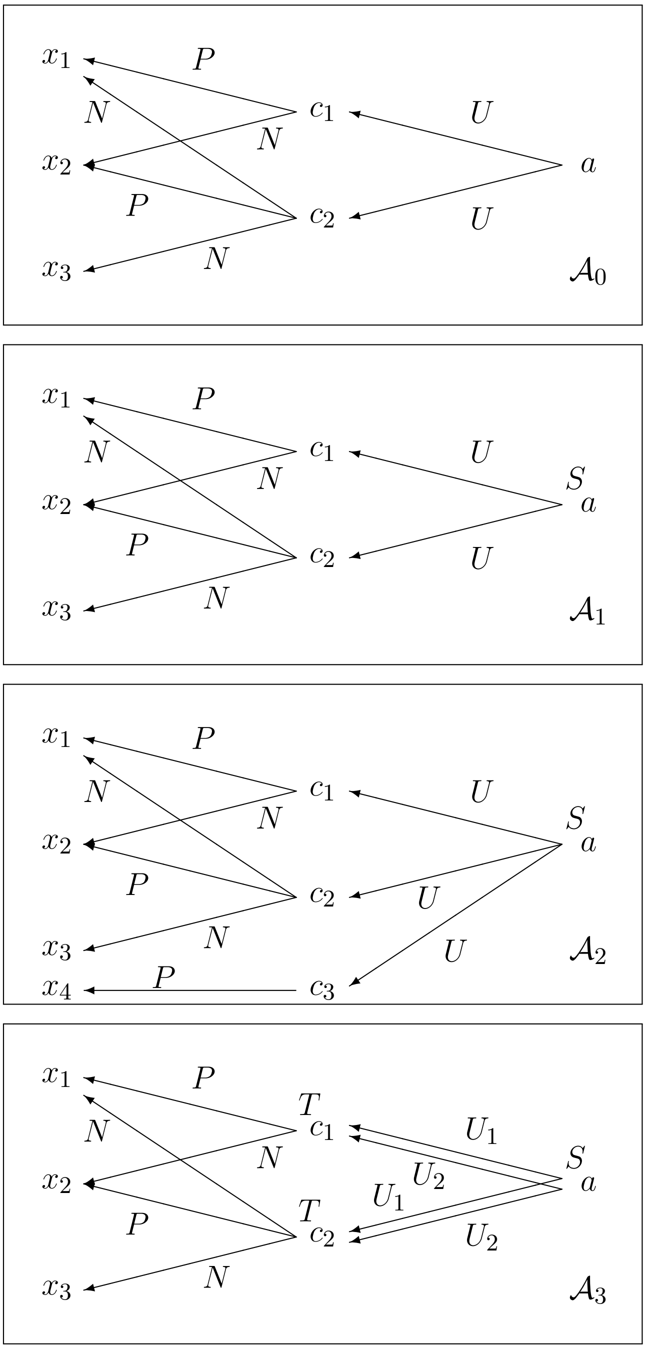
We claim that $\mathcal{E} = \{S(a), T(c_1), ..., T(c_k)\}$ is a smallest explanation for $\langle \mathcal{T}_3, \mathcal{A}_3 \rangle \not\models_{\text{AR}} q_0(a)$ iff φ is unsatisfiable.

$\Rightarrow$ If φ is satisfiable, then we can use a satisfying truth assignment to define a consistent set of k P- and N-edges such that every c_j has an outgoing edge. This set is an explanation for $\langle \mathcal{T}_3, \mathcal{A}_3 \rangle \not\models_{\text{AR}} q_0(a)$, and it has fewer assertions than $\mathcal{E}$.

$\Leftarrow$ If there exists an explanation of size at most k, it contains necessarily only P- and N-edges, since k assertions (P, N or T) are needed to conflict all U_1, and $S(a)$ is needed as soon as one of the U_1-assertions is conflicted only by a T-assertion. It follows that there exists a consistent set of P- and N-assertions such that every c_j has an outgoing edge, from which we can construct a satisfying assignment for φ.

Note that role inclusions are not needed for the lower bound, we can replace $U_1 \sqsubseteq U, U_2 \sqsubseteq U$ by $\exists P \sqsubseteq \neg \exists U_1^-, \exists N \sqsubseteq \neg \exists U_1^-, \exists P \sqsubseteq \neg \exists U_2^-, \exists N \sqsubseteq \neg \exists U_2^-, \exists U_1 \sqsubseteq A, \exists U_2 \sqsubseteq A$ in the reduction. $\square$

Fig. 4.1 Reductions for hardness of explaining positive and negative AR-answers. Graphical representation of the ABox constructed from an example set of clauses $\varphi = \{C_1 = X_1 \vee \neg X_2, C_2 = \neg X_1 \vee X_2 \vee \neg X_3\}$.



4.2.4 Negative IAR-answers

Similarly to the negative AR-answers explanations, we relate explanations of negative IAR-answers to minimal models of the clause set $\varphi_{\neg q}$. Indeed, to show that an answer is not IAR, it is sufficient to contradict every cause, without the consistency constraint.

Proposition 4.2.13. *A set $\mathcal{E}$ is an explanation (resp. cardinality-minimal explanation) for $\mathcal{K} \not\models_{IAR} q(\vec{a})$ iff $\{x_\alpha \mid \alpha \in \mathcal{E}\}$ is a minimal (resp. cardinality-minimal) model of $\varphi_{\neg q}$.*

Proof. The assertions whose corresponding variables are assigned to true in a valuation that satisfies $\varphi_{\neg q}$ form a subset of the ABox which contradicts every cause, since $\varphi_{\neg q}$ states that for every cause, one conflicting assertion is selected. Thus, the inclusion-minimal (resp. cardinality-minimal) models of $\varphi_{\neg q}$ are precisely the explanations (resp. cardinality-minimal explanations) for negative IAR-answers. $\square$

Importantly, $\varphi_{\neg q}$ does not contain any negative literals, and it is known that for *positive* clause sets, a single minimal model can be computed in P, and the associated relevance problem is also in P. We establish properties of the necessary and relevant assertions that we use to compute them.

Lemma 4.2.14. *An assertion is necessary for explaining $\mathcal{K} \not\models_{IAR} q(\vec{a})$ just in the case that it is the only conflict of some cause for $q(\vec{a})$.*

Proof. An assertion is necessary to contradicts every cause iff it is necessary to contradict one cause. This means that an assertion α is necessary for explaining $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ iff there exists a cause $\mathcal{C}$ for $q(\vec{a})$ such that $\mathsf{confl}(\mathcal{C},\mathcal{K}) = \{\alpha\}$. $\square$

Lemma 4.2.15. *An assertion is relevant for explaining $\mathcal{K} \not\models_{IAR} q(\vec{a})$ just in the case that it is in conflict with a cause $\mathcal{C}$ for $q(\vec{a})$ such that for every other cause $\mathcal{C}'$, if $\mathsf{confl}(\mathcal{C}',\mathcal{K}) \subseteq \mathsf{confl}(\mathcal{C},\mathcal{K})$, then $\alpha \in \mathsf{confl}(\mathcal{C}',\mathcal{K})$.*

Proof. To see why this characterization holds, first note that if α is relevant for $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$, then there is a subset $\mathcal{E} \subseteq \mathcal{A}$ with $\alpha \in \mathcal{E}$ such that every cause of $q(\vec{a})$ is in conflict with some assertion in $\mathcal{E}$, and no proper subset of $\mathcal{E}'$ possesses this property. Since $\mathcal{E}$ is a minimal set of assertions having this property, we know that there is some cause $\mathcal{C}$ that does not conflict with any assertion in $\mathcal{E} \setminus \{\alpha\}$, and so there cannot exist another cause $\mathcal{C}'$ such that $\mathsf{confl}(\mathcal{C}',\mathcal{K}) \subseteq \mathsf{confl}(\mathcal{C},\mathcal{K})$ and $\alpha \notin \mathsf{confl}(\mathcal{C}',\mathcal{K})$. Conversely, let us suppose that the assertion α is in conflict with a cause $\mathcal{C}$ of $q(\vec{a})$ and for every other cause $\mathcal{C}'$, $\mathsf{confl}(\mathcal{C}',\mathcal{K}) \subseteq \mathsf{confl}(\mathcal{C},\mathcal{K})$ implies $\alpha \in \mathsf{confl}(\mathcal{C}',\mathcal{K})$. It follows that for every cause $\mathcal{C}'$ of $q(\vec{a})$, either $\alpha \in \mathsf{confl}(\mathcal{C}',\mathcal{K})$, or there exists an assertion $\beta_{\mathcal{C}'} \in \mathsf{confl}(\mathcal{C}',\mathcal{K})$ such that $\beta_{\mathcal{C}'} \notin \mathsf{confl}(\mathcal{C},\mathcal{K})$. We can therefore construct an explanation for $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ by taking α together with some of the assertions $\beta_{\mathcal{C}'}$. $\square$

We next establish the complexity upper bounds.

Proposition 4.2.16. *Regarding explanations for negative IAR-answers,* REC *is in* P*,* BEST REC *is in co*NP*,* NEC *is in* P*,* REL *is in* P*, and* GENONE *is in* P *w.r.t. data complexity.*

Proof. It follows from Definition 4.1.10 and from the fact that in DL-Lite$_{\mathcal{R}}$ conflicts are binary that deciding whether $\mathcal{E} \subseteq \mathcal{A}$ is an explanation for $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ can be done in P (data complexity) by checking:

- for every $\mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})$, inconsistency of $\langle \mathcal{T}, \mathcal{C} \cup \{\alpha\} \rangle$ for some assertion $\alpha \in \mathcal{E}$
- minimality of $\mathcal{E}$: no proper subset $\mathcal{E}' \subsetneq \mathcal{E}$ satisfies the previous condition.

We can decide in NP that an explanation $\mathcal{E}$ is *not* a best explanation (according to some polynomial-time ranking criterion) by guessing a subset $\mathcal{E}' \subseteq \mathcal{A}$ and verifying in P w.r.t. data complexity that $\mathcal{E}'$ is an explanation and that it is better than $\mathcal{E}$ according to the given criterion. This yields a coNP upper bound for BEST REC.

Since causes and conflicts can be computed in P, it follows from Lemma 4.2.15 that deciding whether an assertion is necessary can be done in P.

For REL and GENONE, we can use Proposition 4.2.13 to polynomially reduce these problems to the corresponding problems for minimal models of monotone CNF formulas and exploit known results for that setting. Here we describe polytime procedures for the REL and GENONE that are based upon standard techniques from the propositional setting.

The polynomial upper bound for REL follows from the condition of Lemma 4.2.15, which can be checked in polynomial time by examining the causes and conflicts (which are known to be computable in P w.r.t. data complexity).

For GENONE, we first compute (in P) the set of causes of q and conflicts of $\mathcal{K}$. If there is some cause that does not participate in any conflict, then $\mathcal{K} \models_{\text{IAR}} q(\vec{a})$, so we return 'no'. Otherwise, for each cause $\mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})$, we choose some assertion $\alpha_{\mathcal{C}}$ such that $\alpha_{\mathcal{C}}$ conflicts with some assertion in $\mathcal{C}$. By construction, $\{\alpha_{\mathcal{C}} \mid \mathcal{C} \in \mathsf{causes}(q(\vec{a}), \mathcal{K})\}$ contradicts all causes, which means that this set contains at least one explanation. We therefore proceed to remove one assertion at a time as long the set retains the property of contradicting all causes. When it is no longer possible to remove any assertions, we return the current set of assertions, which is an explanation. $\square$

Finally, we establish the intractability of BEST REC and GENBEST.

Proposition 4.2.17. *Regarding explanations for negative IAR-answers in the case where explanations are ranked by cardinality,* GENBEST *is* NP*-hard, and* BEST REC *is co*NP*-hard w.r.t. data complexity.*

Proof. We give a reduction from the problem of deciding if a truth assignment that satisfies a monotone 2-SAT formula assigns a smallest number of variables to true. This problem is coNP-complete (coNP-hardness can be shown by a straightforward reduction from the complement of the well-known NP-complete vertex cover problem).

Let $\varphi = C_1 \wedge ... \wedge C_k$ be a monotone 2-CNF over the variables $\{X_1, ..., X_n\}$, and let ν be a truth assignment that satisfies φ. Consider the following KB:

$$\begin{aligned}
\mathcal{T} &= \{\exists P_r^- \sqsubseteq \neg T \mid 1 \leq r \leq 2\} \\
\mathcal{A} &= \{T(x_i) \mid 1 \leq i \leq n\} \cup \{P_r(c_j, x_i) \mid X_i \ r^{th} \text{ term of } C_j\} \\
q &= \exists y z_1 z_2 \, P_1(y, z_1) \wedge P_2(y, z_2)
\end{aligned}$$

Fig. 4.2 Reduction for hardness of generating and recognizing best explanations of negative IAR-answers. Graphical representation of the ABox constructed from an example set of clauses $\varphi = \{C_1 = X_1 \vee X_2, C_2 = X_2 \vee X_3\}$.

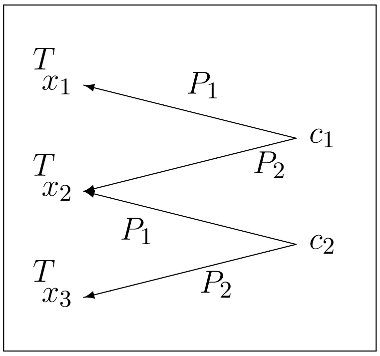


Figure 4.2 illustrates this reduction on an example. The causes for q take the form $\{P_1(c_j, x_{i_1}),\ P_2(c_j, x_{i_2})\}$. It follows that an explanation for $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{\text{IAR}} q$ is a set $\mathcal{E}$ of T-assertions such that for every c_j, there is at least one $X_i \in C_j$ such that $T(x_i) \in \mathcal{E}$.

Deciding if ν assigns a minimal number of variables to true is equivalent to deciding if $\mathcal{E} = \{T(x_i) \mid \nu(X_i) = \mathsf{true}\}$ is a smallest explanation. This yields the coNP-hardness of BEST REC, as well as the NP-hardness of GENBEST: we can solve the minimum assignment problem - and its complement - by generating a cardinality-minimal explanation and comparing its size with the number of variables set to true by the candidate assignment. $\square$

4.3 Implementation and experiments

4.3.1 The explanations framework within CQAPri

To explain why a query answer $\vec{a}$ belongs to one of the three classes Possible, Likely and Sure that correspond to $\mathcal{K} \models_{\text{S}} q(\vec{a})$ and $\mathcal{K} \not\models_{\text{S'}} q(\vec{a})$ for two semantics S and S', CQAPri provides *all* the explanations for $\vec{a}$ being a positive answer under the first semantics and a *single* explanation for it being a negative answer under the other one (i.e. a counter-example), together with the necessary and relevant assertions. For Possible answers, we provide also the necessary and relevant assertions for explaining $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$. Positive explanations are ranked as explained in Section 4.1: using the number of assertions for negative answers and positive brave and IAR-answers, and numbers of disjuncts and total number of assertions for AR-answers; for ranking the later, the user can choose the priority between the two criteria.

Explanations are computed using the results on positive and negative answers from Section 4.2. We thus need the causes of the query answers as well as their conflicts. For the causes, CQAPri prunes the non-minimal images computed during the query answering phase. The conflicts are directly available from the previous steps.

For positive IAR-answers, CQAPri stores the causes without conflict during the query answering time. Instead of halting at the first cause without conflict, it reviews all causes. For positive AR-answers, the SAT encoding is constructed for the query answering phase and CQAPri uses the solver SAT4J to compute the MUSes. Necessary and relevant assertions for positive answers are simply the intersection and union of the explanations. For negative AR-answers, we rely on SAT4J to compute a smallest model of $\varphi_{\neg q} \wedge \varphi_{cons}$, as well as the necessary and relevant assertions with the encodings presented in Propositions 4.2.10 and 4.2.11 that we use to test every potentially relevant assertion, i.e. that appears in $\varphi_{\neg q}$. For negative IAR-answer, we choose to compute by default an arbitrary explanation in polynomial time (cf. Section 4.3.3 for the reason of this choice), but CQAPri can also provide a smallest explanation using the SAT solver to find a cardinality-minimal model of $\varphi_{\neg q}$. The relevant and necessary assertions are computed using Algorithm 4.5 that exploits Lemmas 4.2.14 and 4.2.15.

Algorithm 4.5 RelNecNegIAR

Input: the conflicts of the causes of $q(\vec{a})$: $\mathsf{confl}(\mathcal{C}_1,\mathcal{K}),...,\mathsf{confl}(\mathcal{C}_n,\mathcal{K})$
Output: relevant and necessary assertions for explaining $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$

for all $\mathcal{C}_i$ **do**
  if $|\mathsf{confl}(\mathcal{C}_i,\mathcal{K})| = 1$ **then**
    $Necessary \leftarrow Necessary \cup \mathsf{confl}(\mathcal{C}_i,\mathcal{K})$
  end if
  $Relevant_i \leftarrow \mathsf{confl}(\mathcal{C}_i,\mathcal{K})$
  for all $\mathcal{C}_j$ **do**
    if $\mathsf{confl}(\mathcal{C}_j,\mathcal{K}) \setminus \mathsf{confl}(\mathcal{C}_i,\mathcal{K}) = \emptyset$ **then**
      $Relevant_i \leftarrow Relevant_i \cap \mathsf{confl}(\mathcal{C}_j,\mathcal{K})$
    end if
  end for
end for
$Relevant \leftarrow \bigcup Relevant_i$
Output $Relevant, Necessary$

4.3.2 Experimental setting

To assess the practical interest of our framework, we empirically study the properties of our implementation, in particular: the impact of varying the percentage of assertions in conflict, the typical number and size of explanations, and the extra effort required to generate cardinality-minimal explanations for negative IAR-answers rather than arbitrary ones.

We use the benchmark and experimental setting presented in the Chapter 3 and explain all answers of the queries over all the ABoxes of our benchmark, except for those which have more than 200,000 answers, because it yields unreasonable experimental times.

4.3.3 Experimental results

We summarize below the general tendencies we observed. Table 4.2 shows the number of answers from each class for each query, as well as the distribution of the explanation times

Table 4.2 Number of answers of each class with distribution (in %) of their explanation times (in ms) per query over ABoxes of three different sizes and ratios of conflicts.

		u1c1	#ans	<10	[10,100[	[100,1000[	>1000	**u1c20**	#ans	<10	[10,100[	[100,1000[	>1000	**u1c50**	#ans	<10	[10,100[	[100,1000[	>1000
q1	Sure		20029	100	0	0	0		12538	100	0	0	0		6646	100	0	0	0
	Likely		0						7	100	0	0	0		19	100	0	0	0
	Poss.		380	100	0	0	0		7864	100	0	0	0		13747	99.96	0.04	0	0
q2	Sure		7215	100	0	0	0		6284	100	0	0	0		4728	100	0	0	0
	Likely		20	100	0	0	0		402	99.75	0.25	0	0		887	99.89	0.11	0	0
	Poss.		12	100	0	0	0		734	100	0	0	0		2087	99.95	0.05	0	0
q3	Sure		85	100	0	0	0		0						0				
	Poss.		0						85	100	0	0	0		87	100	0	0	0
q4	Sure		78101	100	0	0	0		24545	100	0	0	0		4806	100	0	0	0
	Poss.		5636	99.96	0.04	0	0		60236	99.99	0.01	0	0		80839	99.98	0.01	<0.01	0
q5	Sure		10	100	0	0	0		0						0				
	Likely		0						10	60	40	0	0		0				
	Poss.		0						0						10	100	0	0	0
q6	Sure		235	100	0	0	0		177	100	0	0	0		0				
	Likely		0						14	100	0	0	0		0				
	Poss.		0						110	100	0	0	0		342	100	0	0	0
q7	Sure		136	100	0	0	0		0						0				
	Poss.		1	100	0	0	0		138	100	0	0	0		149	100	0	0	0
q9	Sure		1291	100	0	0	0		1002	99.99	0.01	0	0		783	100	0	0	0
	Likely		3	100	0	0	0		68	100	0	0	0		116	100	0	0	0
	Poss.		80	100	0	0	0		406	100	0	0	0		741	99.87	0.13	0	0
	Poss.		3	100	0	0	0		6	100	0	0	0		7	100	0	0	0
q11	Sure		534	100	0	0	0		471	100	0	0	0		385	100	0	0	0
	Likely		0						4	100	0	0	0		7	100	0	0	0
	Poss.		4	100	0	0	0		89	100	0	0	0		236	99.58	0.42	0	0
q12	Sure		1180	100	0	0	0		999	100	0	0	0		802	100	0	0	0
	Likely		11	100	0	0	0		174	100	0	0	0		345	100	0	0	0
	Poss.		10	40	20	40	0		117	69.23	4.27	26.50	0		350	73.43	0	26.57	0
q13	Sure		1069	100	0	0	0		966	100	0	0	0		783	100	0	0	0
	Likely		3	100	0	0	0		71	100	0	0	0		169	100	0	0	0
	Poss.		8	100	0	0	0		122	99.18	0	0.82	0		351	98.58	0.28	1.14	0
q14	Sure		191	100	0	0	0		98	100	0	0	0		36	100	0	0	0
	Poss.		4	100	0	0	0		97	100	0	0	0		159	100	0	0	0
q15	Sure		405	100	0	0	0		99	100	0	0	0		0				
	Poss.		102	100	0	0	0		409	99.76	0	0.24	0		515	99.03	0	0.97	0
q16	Sure		13545	99.99	0.01	0	0		2052	100	0	0	0		0				
	Poss.		3987	100	0	0	0		15480	99.99	0.01	0	0		17532	99.98	0.02	0	0
	Poss.		1	100	0	0	0		1	100	0	0	0		1	100	0	0	0
q18	Sure		3107	100	0	0	0		2302	100	0	0	0		1319	100	0	0	0
	Poss.		66	100	0	0	0		872	99.89	0.11	0	0		1871	100	0	0	0
q20	Sure		50	100	0	0	0		25	100	0	0	0		0				
	Poss.		0						25	100	0	0	0		50	100	0	0	0

		u20c1 #ans	<10	[10,100[	[100,1000[	>1000	**u20c20** #ans	<10	[10,100[	[100,1000[	>1000	**u20c50** #ans	<10	[10,100[	[100,1000[	>1000
q2	Sure	189186	>99.99	<0.01	0	0	163260	>99.99	<0.01	0	0	127098	>99.99	<0.01	0	0
	Likely	228	100	0	0	0	9927	99.98	0.02	0	0	23819	99.97	0.03	0	0
	Poss.	1019	100	0	0	0	22489	99.96	0.04	0	0	52117	99.95	0.05	0	0
q3	Sure	85	100	0	0	0	0					0				
	Poss.	0					85	100	0	0	0	87	100	0	0	0
q5	Sure	10	100	0	0	0	0					0				
	Likely	0					10	70	30	0	0	0				
	Poss.	0					0					10	90	10	0	0
q6	Sure	235	100	0	0	0	177	100	0	0	0	0				
	Likely	0					14	100	0	0	0	0				
	Poss.	0					110	100	0	0	0	342	100	0	0	0
q7	Sure	91	100	0	0	0	0					0				
	Poss.	46	100	0	0	0	138	100	0	0	0	149	100	0	0	0
q8	Poss.	31	100	0	0	0	31	100	0	0	0	32	100	0	0	0
q9	Sure	33433	>99.99	<0.01	0	0	25701	>99.99	<0.01	0	0	21462	99.96	0.04	0	0
	Likely	60	100	0	0	0	59	100	0	0	0	267	99.63	0.37	0	0
	Poss.	2714	100	0	0	0	13282	99.89	0.11	0	0	21419	93.01	6.99	0	0
q10	Poss.	58	100	0	0	0	62	100	0	0	0	66	100	0	0	0
q11	Sure	14331	100	0	0	0	12613	100	0	0	0	10329	100	0	0	0
	Likely	0					42	100	0	0	0	267	100	0	0	0
	Poss.	145	100	0	0	0	2781	99.89	0.11	0	0	6373	99.70	0.30	0	0
q12	Sure	7082	100	0	0	0	5830	100	0	0	0	5395	100	0	0	0
	Likely	218	11.93	88.07	0	0	880	25.68	74.32	0	0	991	58.83	40.06	1.11	0
	Poss.	251	47.41	23.90	28.69	0	3881	50.40	18.50	31.10	0	8769	54.29	11.68	34.03	0
q13	Sure	28891	100	0	0	0	25791	100	0	0	0	21471	100	0	0	0
	Likely	64	100	0	0	0	1780	100	0	0	0	4185	99.95	0.05	0	0
	Poss.	204	98.53	0	1.47	0	4028	98.01	0.05	1.94	0	9430	98.30	0.22	1.48	0
q14	Sure	4785	100	0	0	0	2539	100	0	0	0	1007	100	0	0	0
	Likely	0					0					0				
	Poss.	166	100	0	0	0	2412	100	0	0	0	3944	100	0	0	0
q15	Sure	12050	100	0	0	0	1715	100	0	0	0	54	100	0	0	0
	Poss.	1702	99.82	0	0.18	0	12143	99.28	0.02	0.70	0	13946	98.69	0.03	1.28	0
q17	Sure	27	100	0	0	0	0					0				
	Poss.	10	100	0	0	0	37	100	0	0	0	39	82.05	17.95	0	0
q18	Sure	81760	>99.99	<0.01	0	0	58294	99.99	0.01	0	0	34795	99.98	0.02	0	0
	Poss.	1342	100	0	0	0	24959	99.87	0.13	0	0	48770	99.91	0.09	0	0
q19	Poss.	1	0	100	0	0	8	0	87.50	12.50	0	20	0	45	55	0
q20	Sure	50	100	0	0	0	25	100	0	0	0	0				
	Poss.	0					25	100	0	0	0	50	100	0	0	0

		u100c1	#ans	<10	[10,100[	[100,1000[	>1000	u100c20	#ans	<10	[10,100[	[100,1000[	>1000	u100c50	#ans	<10	[10,100[	[100,1000[	>1000
q3	Sure		85	100	0	0	0		0						0				
	Poss.		0						85	100	0	0	0		87	98.85	0	0	1.15
q5	Sure		10	100	0	0	0		0						0				
	Likely		0						10	50	50	0	0		0				
	Poss.		0						0						10	0	100	0	0
q6	Sure		235	100	0	0	0		177	100	0	0	0		0				
	Likely		0						14	100	0	0	0		0				
	Poss.		0						110	100	0	0	0		342	100	0	0	0
q7	Sure		34	100	0	0	0		0						0				
	Poss.		103	100	0	0	0		138	99.28	0	0.72	0		149	78.52	20.81	0.67	0
q8	Poss.		187	100	0	0	0		188	100	0	0	0		190	10.53	88.95	0.52	0
q9	Sure		152404	99.99	0.01	0	0		128616	99.98	0.02	0	0		107220	99.99	0.01	0	0
	Likely		110	100	0	0	0		192	100	0	0	0		1300	70.46	29.54	0	0
	Poss.		27107	99.95	0.05	0	0		64820	86.54	13.46	0	0		105450	63.68	32.92	3.25	0.15
q10	Poss.		293	100	0	0	0		310	96.45	3.55	0	0		326	33.13	66.87	0	0
q11	Sure		71756	>99.99	<0.01	0	0		63411	>99.99	0	<0.01	0		51791	>99.99	<0.01	0	0
	Likely		0						192	100	0	0	0		1300	99.92	0.08	0	0
	Poss.		739	100	0	0	0		13778	99.92	0.06	0.02	0		31764	99.76	0.24	0	0
q12	Sure		31955	100	0	0	0		29074	>99.99	<0.01	0	0		27002	100	0	0	0
	Likely		566	0.53	0	99.47	0		1166	86.19	0	5.57	8.23		2849	99.16	0.03	0	0.81
	Poss.		1109	43.28	12.17	44.55	0		18162	52.16	2.31	38.46	7.07		41644	56.85	0.34	30.08	12.73
q13	Sure		144313	>99.99	<0.01	0	0		129083	>99.99	<0.01	0	0		107258	>99.99	<0.01	0	0
	Likely		308	100	0	0	0		8902	99.98	0.02	0	0		21279	99.99	0.01	0	0
	Poss.		1014	98.82	0	1.18	0		19737	98.56	0.06	1.38	0		46553	98.56	0.16	1.28	0
q14	Sure		23330	100	0	0	0		12390	100	0	0	0		4942	100	0	0	0
	Poss.		777	100	0	0	0		11717	100	0	0	0		19165	99.99	0	0	0.01
q15	Sure		61189	100	0	0	0		7693	100	0	0	0		221	100	0	0	0
	Poss.		7584	99.83	0.01	0.16	0		61492	99.52	0.01	0.47	0		69599	98.99	0.01	1.00	0
q17	Sure		28	100	0	0	0		0						0				
	Poss.		190	100	0	0	0		221	81.90	18.10	0	0		226	3.10	95.57	1.33	0
q19	Poss.		5	0	100	0	0		56	0	98.21	1.79	0		124	0	65.32	34.68	0
q20	Sure		50	100	0	0	0		25	100	0	0	0		0				
	Poss.		0						25	100	0	0	0		50	100	0	0	0

Fig. 4.3 Time in seconds for query answers explanation w.r.t. the size of the ABox for three ratios of conflicts (about 4%, 30%, and 45% of assertions involved in some conflict). For readability, the two figures on the right focus on the queries whose explanation times are lower and whose behaviors are thus not visible on the first one.

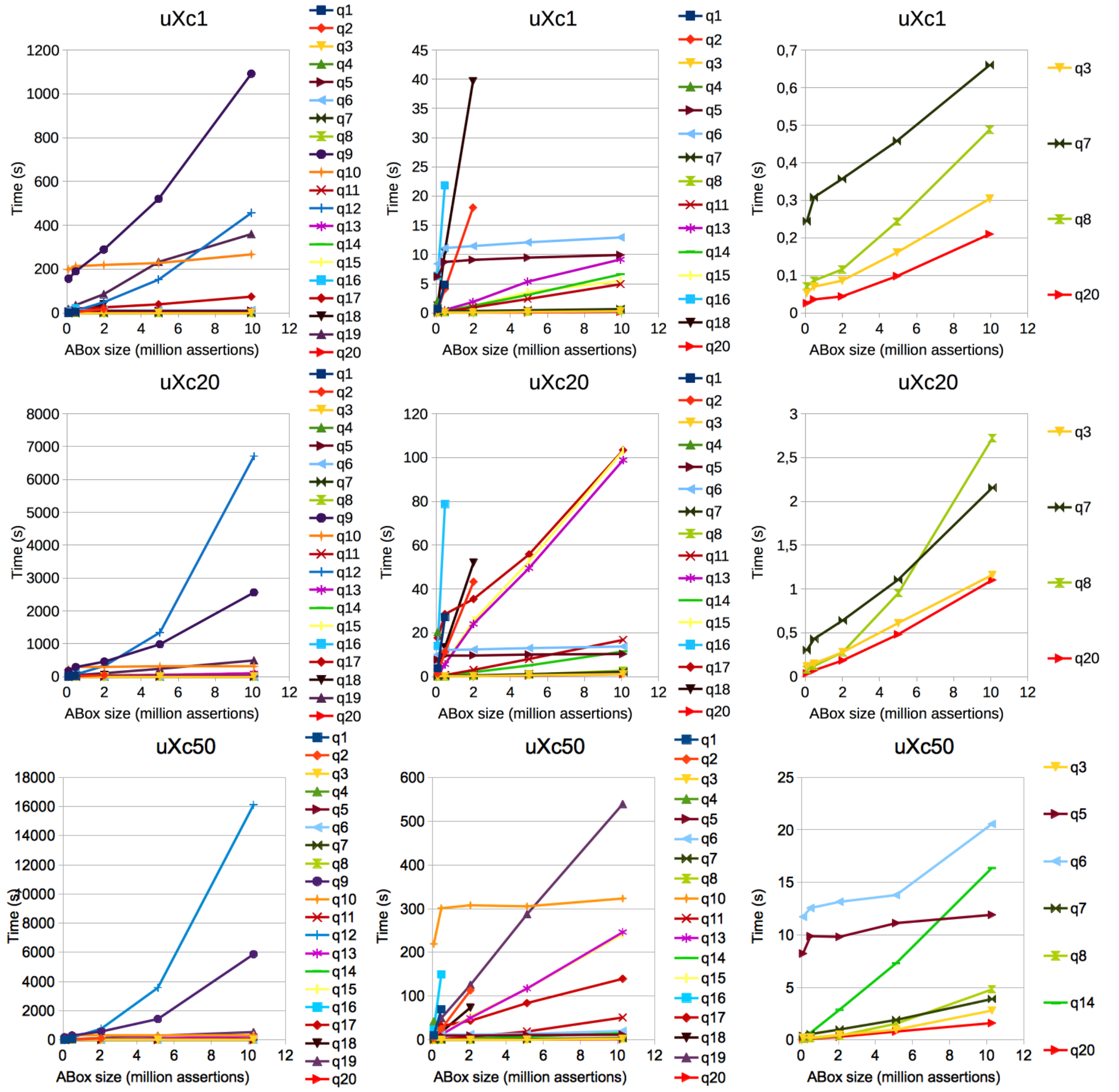


Fig. 4.4 Time in seconds for query answers explanation w.r.t. the ratios of conflicts for three ABox sizes (about 76K, 2 million, and 10 million assertions). For readability, the two figures on the right focus on the queries whose explanation times are lower and whose behaviors are thus not visible on the first one.

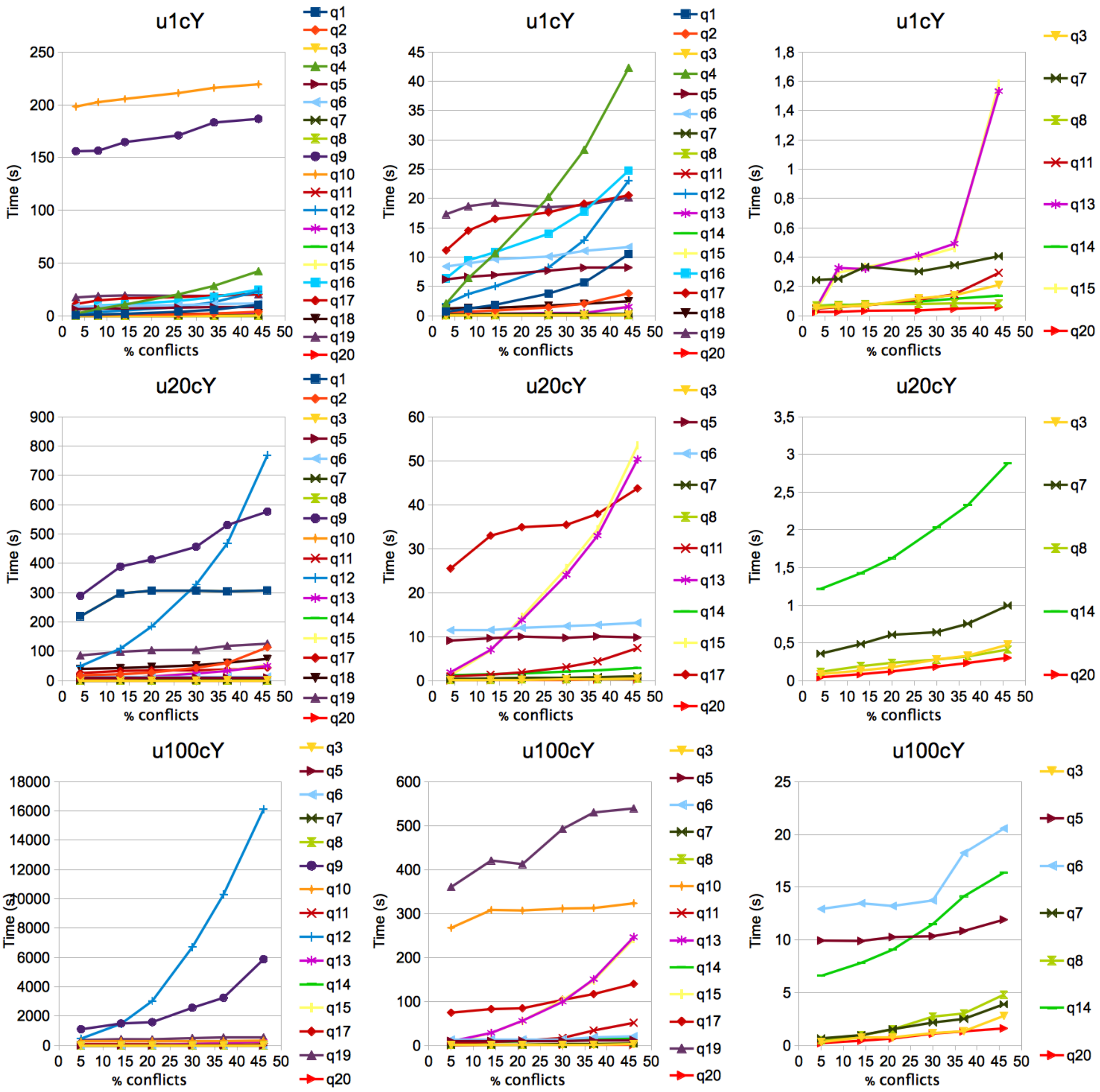
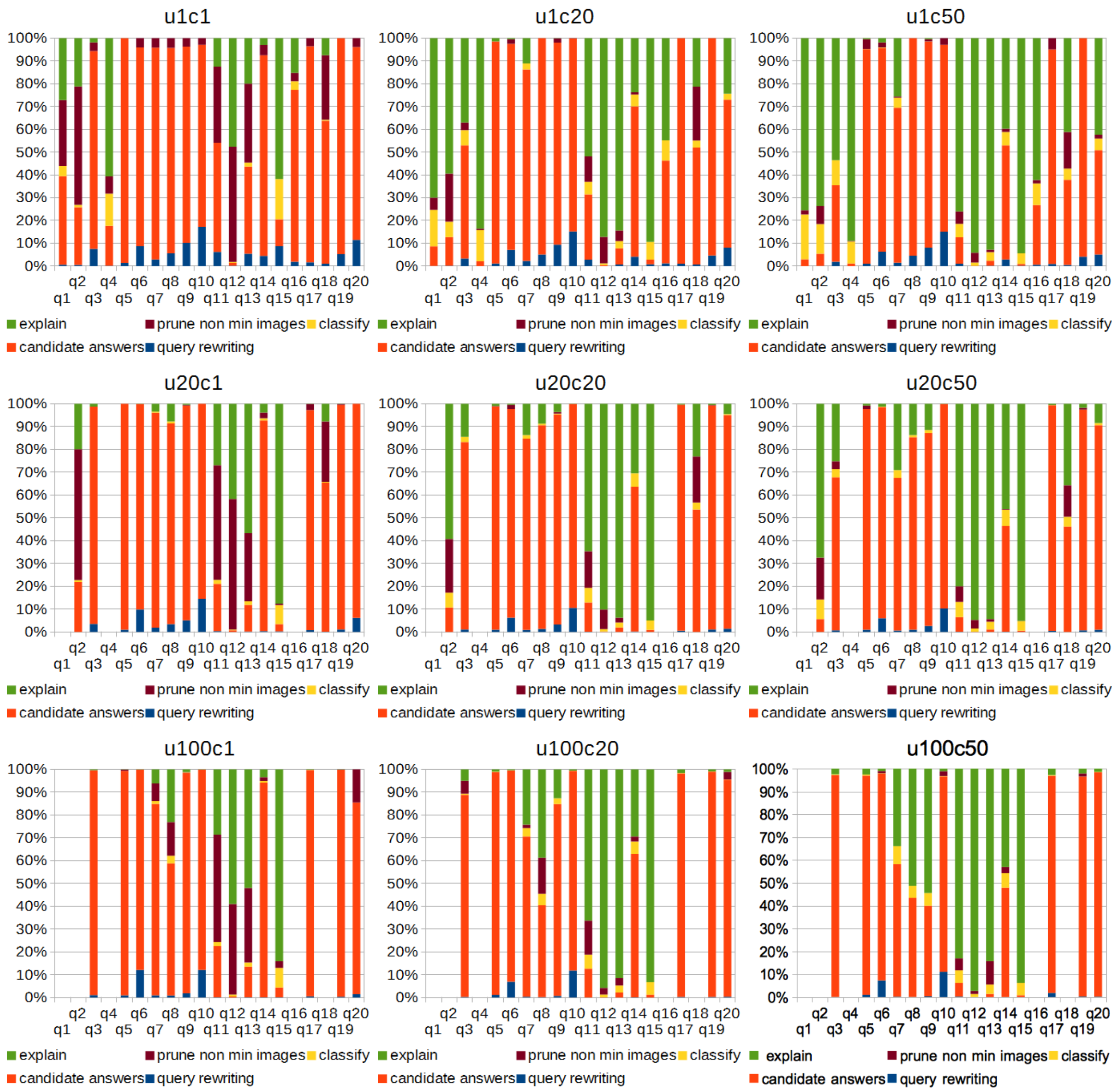


Fig. 4.5 Proportion of time spent by CQAPri in the different phases of query answers explanation on 9 ABoxes: the two lower bars are the time for rewriting the query and executing the rewritten query to get candidate answers, the midle bar is the time needed to classify such answers, and the two upper bars give the total explanation cost, which is divided in the cost of computing the causes by pruning the non-minimal causes and the time needed to compute the explanations from the causes and conflicts.



for these answers, for ABoxes of growing sizes and ratios of conflicts. Figure 4.3 and Figure 4.4 show the time spent in explaining *all* query answers w.r.t. ABox size or proportion of conflicting assertions. Figure 4.5 shows the proportion of time spent in the different phases to explain all query answers for ABoxes of growing difficulty. The explanation cost, given by the two upper bars, consists in pruning non-minimal consistent subsets of the ABox entailing the answers to get the causes, and computing the explanations from the causes and conflicts. The three lower bars relate to the query answering phase, which consists in rewriting and executing the query to get the candidate answers, and identifying Sure, Likely, and Possible answers (classify).

The main conclusion is that explaining a *single* query answer, as described above, is always feasible and fast ($\leq$1s) when there are a *few* percent of conflicts in the ABox (Table 4.2, uXc1 case), as is likely to be the case in most real applications. Even with a *high* percentage of conflicts, the longest time observed is below 20s (19.5s for explaining a Possible answer of q9 on u100c50), and remains lower than 1s for *small* ABoxes (up to u20cY case, i.e. 2 million assertions), and lower than 8s for a *significant* percentage of conflicts (uXc20 case, i.e. 30% of assertions in conflict). In *all* the experiments we made, explaining a *single* answer typically takes less than 10ms, rarely more than 1s. However, computing explanations of *all* answers can be prohibitively expensive when there are very many answers, which is why we do not produce them all by default.

In more detail, adding conflicts to the ABox complicates the explanations of answers, due to their shift from the Sure to the Likely and Possible classes. Explaining such answers indeed comes at higher computational cost. Figures 4.3, 4.4 and 4.5 illustrate this phenomenon. Compared to query answering, explaining all query answers is more sensitive both to ABox size and ratio of conflicts.

We observed that the average number of explanations per answer is often reasonably low, although some answers have a large number of explanations. For instance, on the ABoxes u100cY, we got often less than 10 explanations on average, but this number varies from about 1 to more than 400, and for u100c50, we got up to 4560 for an AR-answer (up to 741 causes for a brave-answer). Even on small ABoxes (u1cY), we got up to 693 explanations for a brave-answer and 243 explanations for an AR-answer. Regarding the size of explanations of AR-answers, the number of causes in the disjunction was up to 25 (for a q12 Likely answer on u100c50; up to 5 on the u1cY ABoxes), showing the practical interest of ranking the explanations.

Explaining negative answers

Our prototype is able to explain $\mathcal{K} \not\models_{S'} q(\vec{a})$ by providing a (possibly smallest) explanation for $\mathcal{K} \not\models_{S'} q(\vec{a})$, together with the relevant and necessary assertions for $\mathcal{K} \not\models_{S'} q(\vec{a})$. We explain here why we chose to compute an arbitrary explanation for $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ by default rather than a cardinality-minimal one. We also give some insight into the explanation times for $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ and the contribution of the computation of necessary and relevant assertions.

We consider the following four cases:

- Case 1 is our default setting, in which we compute an arbitrary explanation for negative IAR-answers, a smallest explanation for negative AR-answers, and the necessary and relevant assertions for explaining negative answers,
- Case 2 differs from Case 1 in omitting the computation of the necessary and relevant assertions for $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ and $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$,
- Case 3 differs from Case 1 in omitting the computation of the relevant assertions for $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$,
- Case 4 differs from Case 1 in computing a cardinality-minimal explanation for $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ instead of using a polynomial-time procedure to generate an arbitrary one.

Table 4.3 displays, for each query, the number of Likely and Possible answers the query possesses, and the distribution of the times for explaining $\mathcal{K} \not\models_{S'} q(\vec{a})$ in our default setting (Case 1). If we compare these distributions with those of Table 4.2, we can see that for many queries, there is the same number of answers having the longest explanation times (columns $[100, 1000[$ and > 1000) when only the negative answer is explained as in the case where both $\mathcal{K} \models_S q(\vec{a})$ and $\mathcal{K} \not\models_{S'} q(\vec{a})$ are explained. This shows that for many queries, the difficulty comes from explaining $\mathcal{K} \not\models_{S'} q(\vec{a})$.

Cost of relevant and necessary assertions Table 4.4 shows the same information as Table 4.3, in the case where the necessary and relevant assertions for explaining $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ or $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ are not computed. In this case, almost all negative answers are explained in less than 10ms. This shows that the main part of the explanation time for negative answers is spent in computing these assertions. Note that for negative IAR-answers (Likely answers), most of the explanation times were already below 10ms in our default case.

Computation of relevant assertions for negative AR-answers Since computing the necessary and relevant assertions for $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ appears to be costly, we investigate further to see how this cost is distributed and if it is possible to reduce the cost of negative explanation without losing too much information. For negative AR-answers, deciding if an assertion α appears in some explanation is an NP-complete problem, and deciding if it appears in all explanations is coNP-complete. In practice the problem of finding the necessary assertions can be solved efficiently because for a negative AR-answer, the SAT solver has already found a model of $\varphi_{\neg q} \cup \varphi_{cons}$ during the query answer classification phase, so checking whether $\varphi_{\neg q} \cup \varphi_{cons} \cup \{\neg x_\alpha\}$ is unsatisfiable is trivial when α does not appear in this model, and if it does, the SAT solver may reuse what it has already computed for a closely related problem. Deciding if α is relevant is more difficult because the encoding we use differs more from $\varphi_{\neg q} \cup \varphi_{cons}$. Indeed, if we compare the execution times with (Table 4.3, for Possible answers) and without (Table 4.5) the computation of the relevant assertions for $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$, we observe significant differences: for all queries and ABoxes, at least 60% of the negative AR-answers are explained in less than 10ms (only 6.45% for q19 on u100c50 in Case 1), and less than 0.15% of them need more than 1s (12.72% of the negative AR-answers of q12

Table 4.3 Distribution of the times (in ms) for explaining $\mathcal{K} \not\models_{\mathrm{S}} q(\vec{a})$ in Case 1.

		u1c1 #ans	<10	[10,100[	[100,1000[	>1000	**u1c20** #ans	<10	[10,100[	[100,1000[	>1000	**u1c50** #ans	<10	[10,100[	[100,1000[	>1000
q1	Likely	0					7	100	0	0	0	19	100	0	0	0
	Poss.	380	100	0	0	0	7864	100	0	0	0	13747	99.96	0.04	0	0
q2	Likely	20	100	0	0	0	402	100	0	0	0	887	100	0	0	0
	Poss.	12	100	0	0	0	734	100	0	0	0	2087	99.95	0.05	0	0
q3	Poss.	0					85	100	0	0	0	87	100	0	0	0
q4	Poss.	5636	99.96	0.04	0	0	60236	100	0	0	0	80839	99.99	<0.01	<0.01	0
q5	Likely	0					10	100	0	0	0	0				
	Poss.	0					0					10	100	0	0	0
q6	Likely	0					14	100	0	0	0	0				
	Poss.	0					110	100	0	0	0	342	100	0	0	0
q7	Poss.	1	100	0	0	0	138	100	0	0	0	149	100	0	0	0
q9	Likely	3	100	0	0	0	68	100	0	0	0	116	100	0	0	0
	Poss.	80	100	0	0	0	406	100	0	0	0	741	99.87	0.13	0	0
q10	Poss.	3	100	0	0	0	6	100	0	0	0	7	100	0	0	0
q11	Likely	0					4	100	0	0	0	7	100	0	0	0
	Poss.	4	100	0	0	0	89	100	0	0	0	236	99.58	0.42	0	0
q12	Likely	11	100	0	0	0	174	100	0	0	0	345	100	0	0	0
	Poss.	10	70	0	30	0	117	79.49	0	20.51	0	350	75.42	2.29	22.29	0
q13	Likely	3	100	0	0	0	71	100	0	0	0	169	100	0	0	0
	Poss.	8	100	0	0	0	122	99.18	0	0.82	0	351	98.58	0.28	1,14	0
q14	Poss.	4	100	0	0	0	97	100	0	0	0	159	100	0	0	0
q15	Poss.	102	100	0	0	0	409	99.76	0	0.24	0	515	99.03	0	0.97	0
q16	Poss.	3987	100	0	0	0	15480	99.99	0.01	0	0	17532	99.99	0.01	0	0
q17	Poss.	1	100	0	0	0	1	100	0	0	0	1	100	0	0	0
q18	Poss.	66	100	0	0	0	872	100	0	0	0	1871	100	0	0	0
q20	Poss.	0					25	100	0	0	0	50	100	0	0	0

		u20c1 #ans	<10	[10,100[	[100,1000[	>1000	**u20c20** #ans	<10	[10,100[	[100,1000[	>1000	**u20c50** #ans	<10	[10,100[	[100,1000[	>1000
q2	Likely	228	100	0	0	0	9927	100	0	0	0	23819	>99.99	<0.01	0	0
	Poss.	1019	100	0	0	0	22489	>99.99	<0.01	0	0	52117	99.97	0.03	0	0
q3	Poss.	0					85	100	0	0	0	87	100	0	0	0
q5	Likely	0					10	100	0	0	0	0				
	Poss.	0					0					10	100	0	0	0
q6	Likely	0					14	100	0	0	0	0				
	Poss.	0					110	100	0	0	0	342	100	0	0	0
q7	Poss.	46	100	0	0	0	138	100	0	0	0	149	100	0	0	0
q8	Poss.	31	100	0	0	0	31	100	0	0	0	32	100	0	0	0
q9	Likely	60	100	0	0	0	59	100	0	0	0	267	100	0	0	0
	Poss.	2714	100	0	0	0	13282	99.92	0.08	0	0	21419	94.47	5.53	0	0
q10	Poss.	58	100	0	0	0	62	100	0	0	0	66	100	0	0	0
q11	Likely	0					42	100	0	0	0	267	100	0	0	0
	Poss.	145	100	0	0	0	2781	99.93	0.07	0	0	6373	99.76	0.24	0	0
q12	Likely	218	100	0	0	0	880	100	0	0	0	991	100	0	0	0
	Poss.	251	66.53	13.55	19.92	0	3881	59.26	17.39	23.35	0	8769	56.43	14.85	28.72	0
q13	Likely	64	100	0	0	0	1780	100	0	0	0	4185	100	0	0	0
	Poss.	204	98.53	0	1.47	0	4028	98.01	0.05	1.94	0	9430	98.35	0.17	1.48	0
q14	Poss.	166	100	0	0	0	2412	100	0	0	0	3944	100	0	0	0
q15	Poss.	1702	99.82	0	0.18	0	12143	99.28	0.02	0.70	0	13946	98.72	0.01	1.27	0
q17	Poss.	10	100	0	0	0	37	100	0	0	0	39	100	0	0	0
q18	Poss.	1342	100	0	0	0	24959	99.97	0.03	0	0	48770	99.99	0.01	0	0
q19	Poss.	1	100	0	0	0	8	75	25	0	0	20	10	90	0	0
q20	Poss.	0					25	100	0	0	0	50	100	0	0	0

		u100c1 #ans	<10	[10,100[	[100,1000[	>1000	**u100c20** #ans	<10	[10,100[	[100,1000[	>1000	**u100c50** #ans	<10	[10,100[	[100,1000[	>1000
q3	Poss.	0					85	100	0	0	0	87	98.85	0	0	1.15
q5	Likely	0					10	100	0	0	0	0				
	Poss.	0					0					10	10	90	0	0
q6	Likely	0					14	100	0	0	0	0				
	Poss.	0					110	100	0	0	0	342	100	0	0	0
q7	Poss.	103	100	0	0	0	138	99.28	0	0.72	0	149	95.30	4.03	0.67	0
q8	Poss.	187	100	0	0	0	188	100	0	0	0	190	42.63	56.84	0.53	0
q9	Likely	110	100	0	0	0	192	100	0	0	0	1300	100	0	0	0
	Poss.	27107	99.99	0.01	0	0	64820	90.70	9.30	0	0	105450	75.55	22.30	2	0.15
q10	Poss.	293	100	0	0	0	310	98.39	1.61	0	0	326	51.23	48.77	0	0
q11	Likely	0					192	100	0	0	0	1300	100	0	0	0
	Poss.	739	100	0	0	0	13778	99.97	0.03	0	0	31764	99.84	0.16	0	0
q12	Likely	566	100	0	0	0	1166	100	0	0	0	2849	100	0	0	0
	Poss.	1109	64.65	0.09	35.26	0	18162	61.67	0.22	31.26	6.84	41644	58.84	2.84	25.60	12.72
q13	Likely	308	100	0	0	0	8902	100	0	0	0	21279	100	0	0	0
	Poss.	1014	98.82	0	1.18	0	19737	98.60	0.02	1.38	0	46553	98.62	0.10	1.28	0
q14	Poss.	777	100	0	0	0	11717	100	0	0	0	19165	99.99	0	0	0.01
q15	Poss.	7584	99.84	0	0.16	0	61492	99.53	<0.01	0.47	0	69599	98.99	0.01	1.00	0
q17	Poss.	190	100	0	0	0	221	95.93	4.07	0	0	226	14.60	84.52	0.88	0
q19	Poss.	5	100	0	0	0	56	91.07	8.93	0	0	124	6.45	93.55	0	0
q20	Poss.	0					25	100	0	0	0	50	100	0	0	0

Table 4.4 Distribution of the times (in ms) for explaining $\mathcal{K} \not\models_{\mathrm{S}} q(\vec{a})$ in Case 2.

	u1c1 #ans	<10	[10,100[	[100,1000[	>1000	u1c20 #ans	<10	[10,100[	[100,1000[	>1000	u1c50 #ans	<10	[10,100[	[100,1000[	>1000
q1 Likely	0					7	100	0	0	0	19	100	0	0	0
Poss.	380	100	0	0	0	7864	100	0	0	0	13747	100	0	0	0
q2 Likely	20	100	0	0	0	402	100	0	0	0	887	100	0	0	0
Poss.	12	100	0	0	0	734	100	0	0	0	2087	100	0	0	0
q3 Poss.	0					85	100	0	0	0	87	100	0	0	0
q4 Poss.	5636	100	0	0	0	60236	100	0	0	0	80839	100	0	0	0
q5 Likely	0					10	100	0	0	0	0				
Poss.	0					0					10	100	0	0	0
q6 Likely	0					14	100	0	0	0	0				
Poss.	0					110	100	0	0	0	342	100	0	0	0
q7 Poss.	1	100	0	0	0	138	100	0	0	0	149	100	0	0	0
q9 Likely	3	100	0	0	0	68	100	0	0	0	116	100	0	0	0
Poss.	80	100	0	0	0	406	100	0	0	0	741	100	0	0	0
q10 Poss.	3	100	0	0	0	6	100	0	0	0	7	100	0	0	0
q11 Likely	0					4	100	0	0	0	7	100	0	0	0
Poss.	4	100	0	0	0	89	100	0	0	0	236	100	0	0	0
q12 Likely	11	100	0	0	0	174	100	0	0	0	345	100	0	0	0
Poss.	10	100	0	0	0	117	100	0	0	0	350	100	0	0	0
q13 Likely	3	100	0	0	0	71	100	0	0	0	169	100	0	0	0
Poss.	8	100	0	0	0	122	100	0	0	0	351	100	0	0	0
q14 Poss.	4	100	0	0	0	97	100	0	0	0	159	100	0	0	0
q15 Poss.	102	100	0	0	0	409	100	0	0	0	515	100	0	0	0
q16 Poss.	3987	100	0	0	0	15480	100	0	0	0	17532	100	0	0	0
q17 Poss.	1	100	0	0	0	1	100	0	0	0	1	100	0	0	0
q18 Poss.	66	100	0	0	0	872	100	0	0	0	1871	100	0	0	0
q20 Poss.	0					25	100	0	0	0	50	100	0	0	0

	u20c1 #ans	<10	[10,100[	[100,1000[	>1000	u20c20 #ans	<10	[10,100[	[100,1000[	>1000	u20c50 #ans	<10	[10,100[	[100,1000[	>1000
q2 Likely	228	100	0	0	0	9927	100	0	0	0	23819	100	0	0	0
Poss.	1019	100	0	0	0	22489	100	0	0	0	52117	100	0	0	0
q3 Poss.	0					85	100	0	0	0	87	100	0	0	0
q5 Likely	0					10	100	0	0	0	0				
Poss.	0					0					10	100	0	0	0
q6 Likely	0					14	100	0	0	0	0				
Poss.	0					110	100	0	0	0	342	100	0	0	0
q7 Poss.	46	100	0	0	0	138	100	0	0	0	149	100	0	0	0
q8 Poss.	31	100	0	0	0	31	100	0	0	0	32	100	0	0	0
q9 Likely	60	100	0	0	0	59	100	0	0	0	267	100	0	0	0
Poss.	2714	100	0	0	0	13282	100	0	0	0	21419	99.94	0.06	0	0
q10 Poss.	58	100	0	0	0	62	100	0	0	0	66	100	0	0	0
q11 Likely	0					42	100	0	0	0	267	100	0	0	0
Poss.	145	100	0	0	0	2781	100	0	0	0	6373	100	0	0	0
q12 Likely	218	100	0	0	0	880	100	0	0	0	991	100	0	0	0
Poss.	251	100	0	0	0	3881	100	0	0	0	8769	100	0	0	0
q13 Likely	64	100	0	0	0	1780	100	0	0	0	4185	100	0	0	0
Poss.	204	100	0	0	0	4028	100	0	0	0	9430	100	0	0	0
q14 Poss.	166	100	0	0	0	2412	100	0	0	0	3944	100	0	0	0
q15 Poss.	1702	100	0	0	0	12143	100	0	0	0	13946	100	0	0	0
q17 Poss.	10	100	0	0	0	37	100	0	0	0	39	100	0	0	0
q18 Poss.	1342	100	0	0	0	24959	100	0	0	0	48770	100	0	0	0
q19 Poss.	1	100	0	0	0	8	100	0	0	0	20	100	0	0	0
q20 Poss.	0					25	100	0	0	0	50	100	0	0	0

	u100c1 #ans	<10	[10,100[	[100,1000[	>1000	u100c20 #ans	<10	[10,100[	[100,1000[	>1000	u100c50 #ans	<10	[10,100[	[100,1000[	>1000
q3 Poss.	0					85	100	0	0	0	87	100	0	0	0
q5 Likely	0					10	100	0	0	0	0				
Poss.	0					0					10	100	0	0	0
q6 Likely	0					14	100	0	0	0	0				
Poss.	0					110	100	0	0	0	342	100	0	0	0
q7 Poss.	103	100	0	0	0	138	100	0	0	0	149	100	0	0	0
q8 Poss.	187	100	0	0	0	188	100	0	0	0	190	100	0	0	0
q9 Likely	110	100	0	0	0	192	100	0	0	0	1300	100	0	0	0
Poss.	27107	100	0	0	0	64820	97.86	2.14	0	0	105450	93.76	6.07	0.02	0.15
q10 Poss.	293	100	0	0	0	310	100	0	0	0	326	100	0	0	0
q11 Likely	0					192	100	0	0	0	1300	100	0	0	0
Poss.	739	100	0	0	0	13778	100	0	0	0	31764	100	0	0	0
q12 Likely	566	100	0	0	0	1166	100	0	0	0	2849	100	0	0	0
Poss.	1109	100	0	0	0	18162	100	0	0	0	41644	>99.99	<0.01	0	0
q13 Likely	308	100	0	0	0	8902	100	0	0	0	21279	100	0	0	0
Poss.	1014	100	0	0	0	19737	100	0	0	0	46553	100	0	0	0
q14 Poss.	777	100	0	0	0	11717	100	0	0	0	19165	100	0	0	0
q15 Poss.	7584	100	0	0	0	61492	100	0	0	0	69599	100	0	0	0
q17 Poss.	190	100	0	0	0	221	100	0	0	0	226	100	0	0	0
q19 Poss.	5	100	0	0	0	56	100	0	0	0	124	100	0	0	0
q20 Poss.	0					25	100	0	0	0	50	100	0	0	0

Table 4.5 Distribution of the times (in ms) for explaining $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ in Case 3.

	u1c1 #ans	<10	[10,100[	[100,1000[	>1000	**u1c20** #ans	<10	[10,100[	[100,1000[	>1000	**u1c50** #ans	<10	[10,100[	[100,1000[	>1000
q1 Poss.	380	100	0	0	0	7864	100	0	0	0	13747	100	0	0	0
q2 Poss.	12	100	0	0	0	734	100	0	0	0	2087	100	0	0	0
q3 Poss.	0					85	100	0	0	0	87	100	0	0	0
q4 Poss.	5636	100	0	0	0	60236	100	0	0	0	80839	>99.99	<0.01	0	0
q5 Poss.	0					0					10	100	0	0	0
q6 Poss.	0					110	100	0	0	0	342	100	0	0	0
q7 Poss.	1	100	0	0	0	138	100	0	0	0	149	100	0	0	0
q9 Poss.	80	100	0	0	0	406	100	0	0	0	741	100	0	0	0
q10 Poss.	3	100	0	0	0	6	100	0	0	0	7	100	0	0	0
q11 Poss.	4	100	0	0	0	89	100	0	0	0	236	100	0	0	0
q12 Poss.	10	70	20	10	0	117	79.49	16.24	4.27	0	350	77.71	18.29	4.00	0
q13 Poss.	8	100	0	0	0	122	100	0	0	0	351	98.87	0.85	0.28	0
q14 Poss.	4	100	0	0	0	97	100	0	0	0	159	100	0	0	0
q15 Poss.	102	100	0	0	0	409	>99.99	<0.01	0	0	515	99.03	0.78	0.19	0
q16 Poss.	3987	100	0	0	0	15480	100	0	0	0	17532	>99.99	<0.01	0	0
q17 Poss.	1	100	0	0	0	1	100	0	0	0	1	100	0	0	0
q18 Poss.	66	100	0	0	0	872	100	0	0	0	1871	100	0	0	0
q20 Poss.	0					25	100	0	0	0	50	100	0	0	0

	u20c1 #ans	<10	[10,100[	[100,1000[	>1000	**u20c20** #ans	<10	[10,100[	[100,1000[	>1000	**u20c50** #ans	<10	[10,100[	[100,1000[	>1000
q2 Poss.	1019	100	0	0	0	22489	100	0	0	0	52117	>99.99	<0.01	0	0
q3 Poss.	0					85	100	0	0	0	87	100	0	0	0
q5 Poss.	0					0					10	100	0	0	0
q6 Poss.	0					110	100	0	0	0	342	100	0	0	0
q7 Poss.	46	100	0	0	0	138	100	0	0	0	149	100	0	0	0
q8 Poss.	31	100	0	0	0	31	100	0	0	0	32	100	0	0	0
q9 Poss.	2714	100	0	0	0	13282	99.96	0.04	0	0	21419	99.69	0.31	0	0
q10 Poss.	58	100	0	0	0	62	100	0	0	0	66	100	0	0	0
q11 Poss.	145	100	0	0	0	2781	100	0	0	0	6373	99.98	0.02	0	0
q12 Poss.	251	79.28	13.95	6.77	0	3881	76.04	18.01	5.95	0	8769	71.05	22.98	5.97	0
q13 Poss.	204	98.53	0.98	0.49	0	4028	98.06	1.29	0.65	0	9430	98.51	1.04	0.45	0
q14 Poss.	166	100	0	0	0	2412	100	0	0	0	3944	100	0	0	0
q15 Poss.	1702	99.82	0.12	0.06	0	12143	99.29	0.50	0.21	0	13946	98.72	0.92	0.35	0
q17 Poss.	10	100	0	0	0	37	100	0	0	0	39	100	0	0	0
q18 Poss.	1342	100	0	0	0	24959	100	0	0	0	48770	100	0	0	0
q19 Poss.	1	100	0	0	0	8	100	0	0	0	20	100	0	0	0
q20 Poss.	0					25	100	0	0	0	50	100	0	0	0

	u100c1 #ans	<10	[10,100[	[100,1000[	>1000	**u100c20** #ans	<10	[10,100[	[100,1000[	>1000	**u100c50** #ans	<10	[10,100[	[100,1000[	>1000
q3 Poss.	0					85	100	0	0	0	87	100	0	0	0
q5 Poss.	0					0					10	100	0	0	0
q6 Poss.	0					110	100	0	0	0	342	100	0	0	0
q7 Poss.	103	100	0	0	0	138	100	0	0	0	149	100	0	0	0
q8 Poss.	187	100	0	0	0	188	100	0	0	0	190	100	0	0	0
q9 Poss.	27107	100	0	0	0	64820	95.40	4.60	0	0	105450	93.64	6.13	0.08	0.15
q10 Poss.	293	100	0	0	0	310	100	0	0	0	326	99.69	0.31	0	0
q11 Poss.	739	100	0	0	0	13778	100	0	0	0	31764	99.99	0.01	0	0
q12 Poss.	1109	64.74	29.94	5.32	0	18162	61.88	27.74	10.38	<0.01	41644	61.68	27.59	10.72	0.01
q13 Poss.	1014	98.82	0.88	0.30	0	19737	98.63	0.84	0.53	0	46553	98.71	0.78	0.50	0.01
q14 Poss.	777	100	0	0	0	11717	100	0	0	0	19165	99.99	0	0.01	0
q15 Poss.	7584	99.84	0.12	0.04	0	61492	99.53	0.29	0.18	0	69599	99.00	0.63	0.37	0
q17 Poss.	190	100	0	0	0	221	100	0	0	0	226	100	0	0	0
q19 Poss.	5	100	0	0	0	56	100	0	0	0	124	100	0	0	0
q20 Poss.	0					25	100	0	0	0	50	100	0	0	0

Table 4.6 Distribution of the times (in ms) for explaining $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ in Case 4.

	u1c1 #ans	<10	[10,100[	[100,1000[	>1000	**u1c20** #ans	<10	[10,100[	[100,1000[	>1000	**u1c50** #ans	<10	[10,100[	[100,1000[	>1000
q1 Likely	0					7	100	0	0	0	19	100	0	0	0
q2 Likely	20	100	0	0	0	402	100	0	0	0	887	100	0	0	0
q5 Likely	0					10	100	0	0	0	0				
q6 Likely	0					14	100	0	0	0	0				
q9 Likely	3	100	0	0	0	68	100	0	0	0	116	100	0	0	0
q11 Likely	0					4	100	0	0	0	7	100	0	0	0
q12 Likely	11	100	0	0	0	174	100	0	0	0	345	100	0	0	0
q13 Likely	3	100	0	0	0	71	100	0	0	0	169	100	0	0	0

	u20c1 #ans	<10	[10,100[	[100,1000[	>1000	**u20c20** #ans	<10	[10,100[	[100,1000[	>1000	**u20c50** #ans	<10	[10,100[	[100,1000[	>1000
q2 Likely	228	100	0	0	0	9927	100	0	0	0	23819	99.82	0.18	0	0
q5 Likely	0					10	100	0	0	0	0				
q6 Likely	0					14	100	0	0	0	0				
q9 Likely	60	100	0	0	0	59	100	0	0	0	267	100	0	0	0
q11 Likely	0					42	100	0	0	0	267	100	0	0	0
q12 Likely	218	100	0	0	0	880	>99.99	0	0	<0.01	991	TO	TO	TO	TO
q13 Likely	64	100	0	0	0	1780	100	0	0	0	4185	100	0	0	0

	u100c1 #ans	<10	[10,100[	[100,1000[	>1000	**u100c20** #ans	<10	[10,100[	[100,1000[	>1000	**u100c50** #ans	<10	[10,100[	[100,1000[	>1000
q5 Likely	0					10	100	0	0	0	0				
q6 Likely	0					14	100	0	0	0	0				
q9 Likely	110	100	0	0	0	192	91.15	1.04	0	7.81	1300	99.84	0.08	0	0.08
q11 Likely	0					192	100	0	0	0	1300	100	0	0	0
q12 Likely	566	100	0	0	0	1166	89.28	10.63	0	0.09	2849	TO	TO	TO	TO
q13 Likely	308	100	0	0	0	8902	99.80	0.20	0	0	21279	99.44	0.56	0	0

on u100c50 in Case 1). Moreover, the longest time required to explain an answer in Case 1 is generally significantly longer than in Case 3 (for instance on u100c20, the longest time required to explain an answer is 8s in Case 1, while it is 5s for Case 3).

We therefore tried to obtain an approximation of these assertions that is fast to compute. The assertions relevant for $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ can be computed very quickly and provide a superset of those for $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$, since the explanations for $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ are the consistent explanations for $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$. However, in our experiments, those two sets of assertions differ quite significantly, and when they do the difference may be huge (hundreds of assertions instead of one to four assertions for some answers of q12 on u100c20). When the ABox size and ratio of conflicts increase, the proportion of answers having additional relevant assertions for $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ and the difference between the two sets of relevant assertions for $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ and $\mathcal{K} \not\models_{\text{AR}} q(\vec{a})$ increase. The two sets always coincide on u1c1, while on the uXc50 ABoxes, they differ in up to 100% of the Possible answers of a query (up to 27 assertions for u1c50, up to 651 assertions for u100c50). On u100c1 they differ in up to 5.6% of the Possible answers of a query, and up to 501 assertions. This shows that it is not possible to gain time on explaining negative AR-answers without losing too much information.

Cardinality of explanations of negative IAR-answers Although smallest explanations for negative answers are preferable, we found it worthwhile to use a polynomial-time method to obtain an arbitrary explanation for a negative IAR-answer rather than relying on the SAT solver to generate a smallest such explanation.

Indeed, computing an arbitrary explanation for $\mathcal{K} \not\models_{\text{IAR}} q(\vec{a})$ always takes less than 100ms (Table 4.3, Likely answers), and in almost all cases less than 10ms.

By contrast, when a *smallest* explanation is computed (Table 4.6), more time may be needed. A striking case is that of q12: on u20c20, almost 19 minutes are spent in computing a smallest explanation for *one* negative IAR-answer, and on u100c20, it takes around 50 minutes, while computing an arbitrary explanation was done in less than 10ms for every ABox and negative IAR-answer of q12 (Table 4.3). This even leads to a time-out for ABoxes from u20c30. This long explanation time is due to the unusual size of the explanation (18 assertions for the answer on u100c20, whereas other negative explanations typically contained only a few assertions).

In terms of size of explanations, we found that on u100c20 the arbitrary explanations generated for all negative IAR-answers of q5, q6, q11, and q12 have exactly the same size as the smallest explanations found with the SAT solver, that only one negative IAR-answer of q13 had a suboptimal explanation (with 4 assertions instead of 3), as well as about 61% of the negative IAR-answers of q9, whose explanations are at most two assertions bigger than the smallest ones.

The possibly very high additional cost of computing a smallest explanation for a limited benefit in terms of the size of explanations lead us to adopt arbitrary explanations for negative IAR-answers as the default setting in our system. However, note that this very high additional cost concerns very few answers, for instance all the other negative IAR-answers of q12 on u20c20 are explained in less than 10ms. It could therefore be possible to allow a short time to first try to find a smallest explanation, and to provide an arbitrary explanation in case of failure.

4.4 Discussion about the notion of responsibility

In the database arena, the notion of *responsibility* has been introduced to quantify the importance of a tuple in the obtention of a (non)answer and can be used to order the tuples of the explanation of this (non)answer [Meliou *et al.* 2010]. The responsibility of a tuple α is a number whose main property is that it is not null just in the case that α is relevant to explain the (non)answer, and equal to one just in the case that α is necessary. Expressed in the DL terminology (in the consistent case and for a positive answer), the responsibility of an assertion α is $\frac{1}{1+k}$ where k is the least number of assertions to delete to make α critical in the obtention of the answer, i.e. to make α belong to every image of the answer.

We tried to adapt this notion to our context and found that it extends well for positive brave and IAR-answers and seems also to be useful for negative IAR-answers, but that we lost the natural intuition for positive and negative AR-answers. The notion of responsibility is based on *contingency sets* which are in the consistent case sets of assertions whose removal makes α critical. For the inconsistency-tolerant semantics we consider, we define contingency sets so that the responsibility fulfills the same main property as the responsibility defined for the consistent case. The following definitions and propositions hold for a KB expressed in any DL. We say that a subset $\mathcal{B}$ of the ABox *contradicts a cause* $\mathcal{C}$ if there exists a consistent subset $\mathcal{B}' \subseteq \mathcal{B}$ such that $\mathcal{C} \cup \mathcal{B}'$ is inconsistent.

Definition 4.4.1 (Contingency set)**.** A contingency set Γ for an assertion α w.r.t. $\langle \mathcal{T}, \mathcal{A} \rangle \models_S q$ is a set of assertions of $\mathcal{A}$ such that:

- For S=IAR:
  - $\langle\mathcal{T},\mathcal{R}_\cap\backslash\Gamma\rangle\models q$ where $\mathcal{R}_\cap=\bigcap_{\mathcal{R}\in Rep(\mathcal{T},\mathcal{A})}\mathcal{R}$
  - $\langle\mathcal{T},(\mathcal{R}_\cap\backslash\Gamma)\backslash\{\alpha\}\rangle\not\models q$
- For S=brave:
  - $\exists\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}),\langle\mathcal{T},\mathcal{R}\backslash\Gamma\rangle\models q$
  - $\forall\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}),\langle\mathcal{T},(\mathcal{R}\backslash\Gamma)\backslash\{\alpha\}\rangle\not\models q$
- For S=AR:
  - $\forall\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}),\langle\mathcal{T},\mathcal{R}\backslash\Gamma\rangle\models q$
  - $\exists\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}),\langle\mathcal{T},(\mathcal{R}\backslash\Gamma)\backslash\{\alpha\}\rangle\not\models q$

A contingency set Γ for an assertion α w.r.t. $\langle\mathcal{T},\mathcal{A}\rangle\not\models_{\mathrm{S}} q$ is a set of assertions of $\mathcal{A}$ such that:

- For S=IAR:
  - $\mathcal{A}\backslash\Gamma$ contradicts every $\mathcal{C}\in\mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\rangle)$
  - $(\mathcal{A}\backslash\Gamma)\backslash\{\alpha\}$ does not contradict every $\mathcal{C}\in\mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\rangle)$
- For S=AR:
  - $\exists\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}),\mathcal{R}\backslash\Gamma$ contradicts every $\mathcal{C}\in\mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\rangle)$
  - $\forall\mathcal{R}\in Rep(\mathcal{T},\mathcal{A}),(\mathcal{R}\backslash\Gamma)\backslash\{\alpha\}$ does not contradict every $\mathcal{C}\in\mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\rangle)$

Definition 4.4.2 (Responsibility)**.** The responsibility of α for $\langle\mathcal{T},\mathcal{A}\rangle\models_{\mathrm{S}} q$ (resp. $\langle\mathcal{T},\mathcal{A}\rangle\not\models_{\mathrm{S'}} q$) is: $\rho(\alpha)=\frac{1}{1+\min_\Gamma|\Gamma|}$, where Γ ranges over all contingency sets for α w.r.t. $\langle\mathcal{T},\mathcal{A}\rangle\models_{\mathrm{S}} q$ (resp. $\langle\mathcal{T},\mathcal{A}\rangle\not\models_{\mathrm{S'}} q$). If there does not exist any contingency set for α, $\rho(\alpha)=0$.

Proposition 4.4.3. *Let $\rho(\alpha)$ be the responsibility of α for $\langle\mathcal{T},\mathcal{A}\rangle\models_S q$ (resp. $\langle\mathcal{T},\mathcal{A}\rangle\not\models_{S'} q$):*

- *$\rho(\alpha)\neq 0$ if and only if α is relevant for explaining $\langle\mathcal{T},\mathcal{A}\rangle\models_S q$ (resp. $\langle\mathcal{T},\mathcal{A}\rangle\not\models_{S'} q$),*
- *$\rho(\alpha)=1$ if and only if α is necessary for explaining $\langle\mathcal{T},\mathcal{A}\rangle\models_S q$ (resp. $\langle\mathcal{T},\mathcal{A}\rangle\not\models_{S'} q$).*

Proof. **Positive IAR-answer**
- Relevance: If α is relevant, there exists a cause $\mathcal{C}_0$ in $\mathcal{R}_\cap$ such that $\alpha\in\mathcal{C}_0$. Then $\Gamma=\bigcup_{\mathcal{C}\in\mathsf{causes}(q,\mathcal{K})}\mathcal{C}\backslash\mathcal{C}_0$ is a contingency set. In the other direction, if $\rho(\alpha)\neq 0$, there exists a contingency set Γ such that every cause in $\mathcal{R}_\cap\backslash\Gamma$ contains α and there exist causes in $\mathcal{R}_\cap\backslash\Gamma$, so there exists a cause which contains α in $\mathcal{R}_\cap$.
- Necessity: α is necessary if and only if α belongs to every cause for q in $\mathcal{R}_\cap$, so if and only if the empty set is a contingency set for α, i.e. $\rho(\alpha)=1$.

Positive brave-answer
- Relevance: If α is relevant, there exists a cause $\mathcal{C}_0$ which contains α. Then $\Gamma=\bigcup_{\mathcal{C}\in\mathsf{causes}(q,\mathcal{K})}\mathcal{C}\backslash\mathcal{C}_0$ is a contingency set. In the other direction, if $\rho(\alpha)\neq 0$, there exists a contingency set Γ such that there exists repair $\mathcal{R}_0$ such that every cause in $\mathcal{R}_0\backslash\Gamma$ contains α and there exists such a cause. It follows that α is relevant.

- Necessity: α is necessary if and only if α belongs to every cause for q, so if and only if the empty set is a contingency set for α, i.e. $\rho(\alpha) = 1$.

Positive AR-answer

- Relevance: If α is relevant, then there exists a minimal disjunction of causes $\mathcal{C}_0$, ..., $\mathcal{C}_n$ such that every repair contains at least one of these causes and $\alpha \in \mathcal{C}_0$. Since the disjunction is minimal, there exists a repair $\mathcal{R}_0$ such that $\mathcal{C}_0 \subseteq \mathcal{R}_0$ and $\mathcal{C}_i \not\subseteq \mathcal{R}_0$ for all $i \neq 0$. Let $\Gamma = \bigcup_{\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})} \mathcal{C} \setminus \bigcup_{i=0}^{n} \mathcal{C}_i$. Since $\mathcal{C}_i \not\subseteq \mathcal{R}_0$ for all $i \neq 0$, $\langle \mathcal{T}, \mathcal{R}_0 \setminus \Gamma \setminus \{\alpha\} \rangle \not\models q$, and since every repair contains at least one $\mathcal{C}_i$, which is disjoint with Γ by construction, $\langle \mathcal{T}, \mathcal{R} \setminus \Gamma \rangle \models q$ for every $\mathcal{R} \in Rep(\mathcal{T}, \mathcal{A})$. It follows that Γ is a contingency set for α, so $\rho(\alpha) \neq 0$.

In the other direction, if $\rho(\alpha) \neq 0$, there exists a contingency set Γ such that $\langle \mathcal{T}, \mathcal{R} \setminus \Gamma \rangle \models q$ for every $\mathcal{R} \in Rep(\mathcal{T}, \mathcal{A})$ and there exists $\mathcal{R}_0 \in Rep(\mathcal{T}, \mathcal{A})$ such that $\langle \mathcal{T}, \mathcal{R}_0 \setminus \Gamma \setminus \{\alpha\} \rangle \not\models q$. Every $\mathcal{R} \setminus \Gamma$ contains at least one cause of q, so it is possible to construct a minimal disjunction of causes with these causes. Since $\langle \mathcal{T}, \mathcal{R}_0 \setminus \Gamma \setminus \{\alpha\} \rangle \not\models q$, every cause in $\mathcal{R}_0 \setminus \Gamma$ contains α, so α appears in the disjunction. This disjunction of causes is such that every repair contains at least one of the causes, and it is minimal (otherwise, the minimal sub-disjunction covers also the $\mathcal{R} \setminus \Gamma$). Hence α is relevant.

- Necessity: If α is necessary, then there exists a repair $\mathcal{R}$ such that every cause included in $\mathcal{R}$ contains α (cf. proof of Proposition 4.2.5). It follows that $\langle \mathcal{T}, \mathcal{R} \setminus \{\alpha\} \rangle \not\models q$, so the empty set is a contingency set for α and $\rho(\alpha) = 1$. In the other direction, if $\rho(\alpha) = 1$, then there exists a repair $\mathcal{R}$ such that $\langle \mathcal{T}, \mathcal{R} \setminus \{\alpha\} \rangle \not\models q$, so every cause for q in $\mathcal{R}$ contains α, and every explanation for $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{AR}} q$ contains α.

Negative IAR-answer

- Relevance: If α is relevant, there exists a minimal subset $\mathcal{B}$ such that $\alpha \in \mathcal{B}$ and $\mathcal{B}$ contradicts every cause. Then $\Gamma = \mathcal{A} \setminus \mathcal{B}$ is a contingency set by construction of $\mathcal{B}$ since $\mathcal{A} \setminus \Gamma = \mathcal{B}$. In the other direction, if there exists Γ as required, $\mathcal{B} = \mathcal{A} \setminus \Gamma$ contradicts every cause, and $\mathcal{B} \setminus \{\alpha\}$ does not contradict every cause, so $\alpha \in \mathcal{B}$. It follows that α is relevant.
- Necessity: It is clear that α is necessary if and only if $\mathcal{A}$ contradicts every cause (q is not entailed under IAR semantics), and $\mathcal{A} \setminus \alpha$ does not contradict at least one cause (every set that contradicts every cause contains α).

Negative AR-answer

- Relevance: If α is relevant, there exists a minimal consistent subset $\mathcal{B}$ such that $\alpha \in \mathcal{B}$ and $\mathcal{B}$ conflicts every cause of q. Since $\mathcal{B}$ is consistent, there exists $\mathcal{R}_0 \in Rep(\mathcal{T}, \mathcal{A})$ such that $\mathcal{B} \subseteq \mathcal{R}_0$. Let $\Gamma = \mathcal{A} \setminus \mathcal{B}$. Then $\mathcal{R}_0 \setminus \Gamma = \mathcal{B}$ contradicts every cause for q, and for every repair $\mathcal{R}$, $\mathcal{R} \setminus \Gamma \setminus \{\alpha\} \subseteq \mathcal{B} \setminus \{\alpha\}$ does not contradict every cause for q since $\mathcal{B}$ is minimal.

In the other direction, suppose that there exists a set Γ as required. Then there exists $\mathcal{R}_0 \in Rep(\mathcal{T}, \mathcal{A})$, such that $\mathcal{R}_0 \setminus \Gamma$ contradicts every cause. Since $\mathcal{R}_0$ is consistent, a negative explanation is obtained by selecting a minimal subset of $\mathcal{R}_0 \setminus \Gamma$ which contradicts every cause. Since $\mathcal{R}_0 \setminus \Gamma \setminus \{\alpha\}$ does not contradict every cause, this negative explanation contains α.

- Necessity: Suppose that α is necessary. Since α belongs to every consistent subset which contradicts the causes of q, for every repair $\mathcal{R}$, $\mathcal{R} \setminus \{\alpha\}$ does not contradict every cause. Since q is not entailed under AR semantics, there exists $\mathcal{R}_0$ such that $\mathcal{R}_0$ contradicts every cause. It follows that the empty set is a contingency set.

In the other direction, suppose that $\rho(\alpha) = 1$ and suppose for a contradiction that there exists a consistent subset $\mathcal{B}$ which contradicts every cause and does not contain α. There exists a repair $\mathcal{R}_0$ such that $\mathcal{B} \subseteq \mathcal{R}_0$, and $\mathcal{R}_0 \backslash \{\alpha\}$ does not contradict the causes for q since the empty set is a contingency set. It follows that $\mathcal{B} \subseteq \mathcal{R}_0 \backslash \{\alpha\}$ does not contradict every cause. $\square$

The following proposition shows that for positive brave or IAR-answers and negative IAR-answers, the responsibility has exactly the same meaning as in the database setting: it corresponds to $\frac{1}{1+k}$ where k is the least number of assertions to delete to make α critical in the (non) obtention of the answer under brave or IAR semantics. In particular, this means that for S $\in$ {IAR, brave}, α is necessary for explaining $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{S}} q$ if and only if $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{S}} q$ and $\langle \mathcal{T}, \mathcal{A} \backslash \{\alpha\} \rangle \not\models_{\text{S}} q$, and that α is necessary for explaining $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{\text{IAR}} q$ if and only if $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{\text{IAR}} q$ and $\langle \mathcal{T}, \mathcal{A} \backslash \{\alpha\} \rangle \models_{\text{IAR}} q$.

Proposition 4.4.4. *If Γ is a contingency set for α w.r.t. $\langle \mathcal{T}, \mathcal{A} \rangle \models_S q$ for S $\in$ {IAR, brave}:*

- $\langle \mathcal{T}, \mathcal{A} \backslash \Gamma \rangle \models_S q$
- $\langle \mathcal{T}, (\mathcal{A} \backslash \Gamma) \backslash \{\alpha\} \rangle \not\models_S q$

If Γ is a minimal *contingency set for α w.r.t. $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{IAR} q$:*

- $\langle \mathcal{T}, \mathcal{A} \backslash \Gamma \rangle \not\models_{IAR} q$
- $\langle \mathcal{T}, (\mathcal{A} \backslash \Gamma) \backslash \{\alpha\} \rangle \models_{IAR} q$

Proof. For positive IAR-answers, by definition of a contingency set $\langle \mathcal{T}, \mathcal{R}_\cap \backslash \Gamma \rangle \models q$ where $\mathcal{R}_\cap = \bigcap_{\mathcal{R} \in Rep(\mathcal{T}, \mathcal{A})} \mathcal{R}$, so there is a cause for q in $\mathcal{R}_\cap \backslash \Gamma \subseteq \bigcap_{\mathcal{R} \in Rep(\mathcal{T}, \mathcal{A} \backslash \Gamma)} \mathcal{R}$ (every assertion in $\mathcal{R}_\cap$ is free of conflicts in $\mathcal{A}$, so also in $\mathcal{A} \backslash \Gamma$). Thus $\langle \mathcal{T}, \mathcal{A} \backslash \Gamma \rangle \models_{\text{IAR}} q$. Moreover, $\langle \mathcal{T}, (\mathcal{R}_\cap \backslash \Gamma) \backslash \{\alpha\} \rangle \not\models q$, so there is no cause for q in $(\mathcal{R}_\cap \backslash \Gamma) \backslash \{\alpha\}$. It follows that α belongs to every cause for q in $\mathcal{R}_\cap \backslash \Gamma$. In particular, this means that α has no conflict, so removing α does not make any assertion of $\mathcal{A}$ conflict free, and every cause for q in $\bigcap_{\mathcal{R} \in Rep(\mathcal{T}, (\mathcal{A} \backslash \Gamma) \backslash \{\alpha\})} \mathcal{R}$ was in $\mathcal{R}_\cap \backslash \Gamma$. It follows that $\bigcap_{\mathcal{R} \in Rep(\mathcal{T}, (\mathcal{A} \backslash \Gamma) \backslash \{\alpha\})} \mathcal{R}$ contains no cause for q.

For positive brave-answers, since there exists $\mathcal{R} \in Rep(\mathcal{T}, \mathcal{A})$ such that $\langle \mathcal{T}, \mathcal{R} \backslash \Gamma \rangle \models q$, there is a cause for q in $\mathcal{R} \backslash \Gamma$ so in $\mathcal{A} \backslash \Gamma$ and $\langle \mathcal{T}, \mathcal{A} \backslash \Gamma \rangle \models_{\text{brave}} q$. Moreover, since for every $\mathcal{R} \in Rep(\mathcal{T}, \mathcal{A})$, $\langle \mathcal{T}, (\mathcal{R} \backslash \Gamma) \backslash \{\alpha\} \rangle \not\models q$, $\Gamma \cup \{\alpha\}$ intersects every cause for q in $\mathcal{A}$, so $\langle \mathcal{T}, (\mathcal{A} \backslash \Gamma) \backslash \{\alpha\} \rangle \not\models_{\text{brave}} q$.

For negative IAR-answers, a minimal contingency set Γ is such that $\mathcal{A} \backslash \Gamma$ contradicts every cause for q, so $\langle \mathcal{T}, \mathcal{A} \backslash \Gamma \rangle \not\models_{\text{IAR}} q$. Moreover, since Γ is a minimal contingency set, there exists a cause $\mathcal{C}_0$ such that $\mathsf{confl}(\mathcal{C}_0, \mathcal{K}) = \Gamma \cup \{\alpha\}$ (indeed, $(\mathcal{A} \backslash \Gamma) \backslash \{\alpha\}$ does not contradict every cause for q, so there exists a cause $\mathcal{C}$ such that $\mathsf{confl}(\mathcal{C}, \mathcal{K}) \subseteq \Gamma \cup \{\alpha\}$, and by minimality of Γ, $\Gamma \cup \{\alpha\}$ corresponds exactly to the conflicts of one such cause). Since $\mathcal{C}_0$ is consistent, $\mathcal{C}_0 \cap (\Gamma \cup \{\alpha\}) = \emptyset$, so $(\mathcal{A} \backslash \Gamma) \backslash \{\alpha\}$ contains $\mathcal{C}_0$ and none of its conflicts, and $\langle \mathcal{T}, (\mathcal{A} \backslash \Gamma) \backslash \{\alpha\} \rangle \models_{\text{IAR}} q$. $\square$

The following examples illustrate how responsibility can help to understand a (positive brave or IAR or negative IAR) answer by ordering the relevant assertions.

Example 4.4.5. Suppose that we have the following KB and query.

$$\begin{aligned}
\mathcal{T} =& \{\mathsf{GradCourse} \sqsubseteq \neg\mathsf{UndergradCourse}\}\\
\mathcal{A} =& \{\mathsf{Postdoc}(ann), \mathsf{Teach}(ann, c_{a1}), \mathsf{GradCourse}(c_{a1}), \mathsf{Teach}(ann, c_{a2}),\\
& \mathsf{GradCourse}(c_{a2}), \mathsf{Teach}(ann, c_{a3}), \mathsf{GradCourse}(c_{a3}),\\
& \mathsf{Postdoc}(bob), \mathsf{Teach}(bob, c_{bc}), \mathsf{Postdoc}(carl), \mathsf{Teach}(carl, c_{bc}), \mathsf{GradCourse}(c_{bc}),\\
& \mathsf{Postdoc}(dan), \mathsf{Teach}(dan, c_d), \mathsf{GradCourse}(c_d), \mathsf{UndergradCourse}(c_d)\}\\
q =& \exists xy\, \mathsf{Postdoc}(x) \wedge \mathsf{Teach}(x, y) \wedge \mathsf{GradCourse}(y)
\end{aligned}$$

If a user wonders why q is entailed under brave semantics (for instance because he thinks that postdoctoral researchers should not teach graduate courses), he will get the 6 following causes as explanations for $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{brave}} q$, and every assertion from $\mathcal{A}$ is relevant except $\mathsf{UndergradCourse}(c_d)$:

- $\{\mathsf{Postdoc}(ann), \mathsf{Teach}(ann, c_{ai}), \mathsf{GradCourse}(c_{ai})\}$ for $1 \leq i \leq 3$,
- $\{\mathsf{Postdoc}(bob), \mathsf{Teach}(bob, c_{bc}), \mathsf{GradCourse}(c_{bc})\}$,
- $\{\mathsf{Postdoc}(carl), \mathsf{Teach}(carl, c_{bc}), \mathsf{GradCourse}(c_{bc})\}$,
- $\{\mathsf{Postdoc}(dan), \mathsf{Teach}(dan, c_d), \mathsf{GradCourse}(c_d)\}$.

Ordering the assertions w.r.t. their responsibility gives the following ranking:

- $\rho = 0.33$ $\mathsf{Postdoc}(ann)$
- $\rho = 0.33$ $\mathsf{GradCourse}(c_{bc})$
- $\rho = 0.33$ $\mathsf{Postdoc}(dan)$
- $\rho = 0.33$ $\mathsf{Teach}(dan, c_d)$
- $\rho = 0.33$ $\mathsf{GradCourse}(c_d)$
- $\rho = 0.25$ $\mathsf{Postdoc}(bob)$
- $\rho = 0.25$ $\mathsf{Teach}(bob, c_{bc})$
- $\rho = 0.25$ $\mathsf{Postdoc}(carl)$
- $\rho = 0.25$ $\mathsf{Teach}(carl, c_{bc})$
- $\rho = 0.20$ $\mathsf{Teach}(ann, c_{a1})$
- $\rho = 0.20$ $\mathsf{GradCourse}(c_{a1})$
- $\rho = 0.20$ $\mathsf{Teach}(ann, c_{a2})$
- $\rho = 0.20$ $\mathsf{GradCourse}(c_{a2})$
- $\rho = 0.20$ $\mathsf{Teach}(ann, c_{a3})$
- $\rho = 0.20$ $\mathsf{GradCourse}(c_{a3})$

If it is the case that postdoctoral researchers do not teach graduate courses:

- either $\mathsf{Postdoc}(ann)$ is erroneous, or at least three assertions are erroneous (either $\mathsf{Teach}(ann, c_{ai})$ or $\mathsf{GradCourse}(c_{ai})$ for each c_{ai});
- either $\mathsf{GradCourse}(c_{bc})$ is erroneous, or at least two others assertions are erroneous (either $\mathsf{Postdoc}(x)$ or $\mathsf{Teach}(x, c_{bc})$ for $x \in \{bob, carl\}$);
- one of the three assertions $\mathsf{Postdoc}(dan)$, $\mathsf{Teach}(dan, c_d)$, and $\mathsf{GradCourse}(c_d)$ is erroneous.

Therefore, if we suppose that every assertion has the same probability of being erroneous a priori, the assertions of responsibility 0.33 are more likely to be problematic than those of lower responsibility, since we have to remove more of the latter to lose the entailment of q. The assertions of higher responsibility are more important in getting the answer. This shows how ranking the relevant assertions by responsibility helps to understand a surprising answer, or to find problematic assertions in case of erroneous brave-answer.

If we now consider IAR semantics, there are 5 explanations for $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{IAR}} q$ (the 5 first causes, so this time $\mathsf{Postdoc}(dan), \mathsf{Teach}(dan, c_d), \mathsf{GradCourse}(c_d)$ are not relevant). The responsibility is as follows:

- $\rho = 0.5$ $\mathsf{Postdoc}(ann)$
- $\rho = 0.5$ $\mathsf{GradCourse}(c_{bc})$
- $\rho = 0.33$ $\mathsf{Postdoc}(bob)$
- $\rho = 0.33$ $\mathsf{Teach}(bob, c_{bc})$
- $\rho = 0.33$ $\mathsf{Postdoc}(carl)$
- $\rho = 0.33$ $\mathsf{Teach}(carl, c_{bc})$
- $\rho = 0.25$ $\mathsf{Teach}(ann, c_{a1})$
- $\rho = 0.25$ $\mathsf{GradCourse}(c_{a1})$
- $\rho = 0.25$ $\mathsf{Teach}(ann, c_{a2})$
- $\rho = 0.25$ $\mathsf{GradCourse}(c_{a2})$
- $\rho = 0.25$ $\mathsf{Teach}(ann, c_{a3})$
- $\rho = 0.25$ $\mathsf{GradCourse}(c_{a3})$

Here again, by ranking assertions using their responsibility, we get first the assertions that play a more important role in deriving the answer under IAR semantics. $\triangleleft$

Example 4.4.6. Regarding negative IAR-answers, consider the following KB and query:

$$\begin{aligned}
\mathcal{T} =& \{\mathsf{AProf} \sqsubseteq \mathsf{Prof}, \mathsf{FProf} \sqsubseteq \mathsf{Prof}, \exists \mathsf{Advise} \sqsubseteq \mathsf{Prof}, \mathsf{Prof} \sqsubseteq \mathsf{PhD}, \mathsf{Postdoc} \sqsubseteq \mathsf{PhD}, \\
& \exists \mathsf{Teach}^- \sqsubseteq \mathsf{Course}, \mathsf{AProf} \sqsubseteq \neg \mathsf{FProf}, \mathsf{Prof} \sqsubseteq \neg \mathsf{Postdoc}, \exists \mathsf{Advise}^- \sqsubseteq \neg \mathsf{Course}\} \\
\mathcal{A} =& \{\mathsf{AProf}(ann), \mathsf{FProf}(ann), \mathsf{Advise}(ann, c), \mathsf{Course}(c), \mathsf{Teach}(ann, c), \mathsf{Postdoc}(ann)\} \\
q =& \mathsf{PhD}(ann)
\end{aligned}$$

The causes of q are $\{\mathsf{AProf}(ann)\}$, $\{\mathsf{FProf}(ann)\}$, $\{\mathsf{Advise}(ann, c)\}$, and $\{\mathsf{Postdoc}(ann)\}$.

There are four explanations for $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{\text{IAR}} q$: $\mathsf{Postdoc}(ann) \wedge \mathsf{Advise}(ann, c)$, $\mathsf{Postdoc}(ann) \wedge \mathsf{AProf}(ann)$, $\mathsf{Postdoc}(ann) \wedge \mathsf{FProf}(ann)$, $\mathsf{Course}(c) \wedge \mathsf{AProf}(ann) \wedge \mathsf{FProf}(ann)$ and $\mathsf{Teach}(ann, c) \wedge \mathsf{AProf}(ann) \wedge \mathsf{FProf}(ann)$.

- $\rho = 0.50$ $\mathsf{AProf}(ann)$
- $\rho = 0.50$ $\mathsf{FProf}(ann)$
- $\rho = 0.50$ $\mathsf{Postdoc}(ann)$
- $\rho = 0.33$ $\mathsf{Advise}(ann, c)$
- $\rho = 0.33$ $\mathsf{Course}(c)$
- $\rho = 0.33$ $\mathsf{Teach}(ann, c)$

Ordering the assertions w.r.t. their responsibility allows us to get first the assertions that contradict causes that have fewer conflicts ($\mathsf{AProf}(ann)$ and $\mathsf{FProf}(ann)$), then those who conflict the causes which have more conflicts ($\mathsf{Postdoc}(ann)$ and $\mathsf{Advise}(ann, c)$). If the query should be entailed under IAR semantics, we can think that the former causes have a higher probability of being correct, and so their conflicts of being erroneous. $\triangleleft$

For positive or negative AR-answers, the responsibility is not related to the minimal number of changes to make α critical in the (non) obtention of the answer under AR semantics. Indeed, it is related to the minimal number of changes to make α critical in the (non) obtention of the answer *in one repair*, but deleting assertions changes the repairs of the knowledge base. For instance if $\mathsf{Prof}(a)$ holds under AR semantics with only one explanation

$\mathsf{AProf}(a) \vee \mathsf{FProf}(a)$, both assertions are necessary to explain that $\mathsf{Prof}(a)$ holds under AR semantics, so have a responsibility of 1, but deleting one of the two necessary assertions will make $\mathsf{Prof}(a)$ be entailed under IAR semantics. This shows that it is not possible to define a notion that is related to the minimal number of changes to make an assertion critical in the obtention of an answer under AR semantics and which is not null just in the case that α is relevant to explain the answer and equal to one just in the case that α is necessary. The following example illustrates the same phenomenon for negative AR-answers.

Example 4.4.7. In this example, we consider a KB defined by a set of assertions $\mathcal{B}$ and conflicts E between them. It is always possible to find a KB which corresponds to such a specification by defining a set of individuals $N_I = \{a\}$, a set of concepts $N_C = \{C \mid C \in \mathcal{B}\}$, the ABox $\mathcal{A} = \{C(a) \mid C \in N_C\}$, and the TBox $\mathcal{T} = \{C_1 \sqsubseteq \neg C_2 \mid \{C_1, C_2\} \in E\}$. We can also assume that a query q has for causes some sets of assertions $\mathcal{C}_1, ..., \mathcal{C}_k$ by defining $q = A_1(a) \wedge ... \wedge A_n(a)$ where n is the maximal size of the $\mathcal{C}_i$, and adding for every $\mathcal{C}_i = \{C_1(a), ..., C_{m_i}(a)\}$ the inclusions $C_1 \sqsubseteq A_1, ..., C_{m_i} \sqsubseteq A_{m_i}, ..., C_{m_i} \sqsubseteq A_n$ to the TBox.

Let $\mathcal{A} = \{\alpha, \beta, \gamma, \delta, \epsilon\}$ and $\mathcal{T}$ be such that $\langle \mathcal{A}, \mathcal{T} \rangle$ has the following conflicts: $\{\alpha, \epsilon\}$, $\{\beta, \gamma\}$, $\{\gamma, \delta\}$, $\{\delta, \epsilon\}$. Suppose that a query q has two causes: $\mathcal{C}_0 = \{\alpha, \beta\}$, $\mathcal{C}_1 = \{\epsilon\}$. Then q is not entailed under AR semantics and there is only one explanation for $\langle \mathcal{A}, \mathcal{T} \rangle \not\models_{\text{AR}} q$: $\{\alpha, \gamma\}$. If we remove the necessary assertion α, since the remaining cause $\mathcal{C}_1$ can be contradicted by δ, q is still not entailed under AR semantics. Therefore, even if $\rho(\alpha) = 1$, α is not critical in the obtention of q. By contrast, γ is critical since the only way of contradicting $\mathcal{C}_0$ in $\mathcal{A} \backslash \{\gamma\}$ is ϵ, so no consistent subset contradicts both $\mathcal{C}_0$ and $\mathcal{C}_1$. $\triangleleft$

5

QUERY-DRIVEN REPAIRING

In this chapter, we address the problem of query-driven repairing of inconsistent DL-Lite$_{\mathcal{R}}$ knowledge bases: query answers are computed under inconsistency-tolerant semantics, and the user provides feedback about which answers are erroneous, or missing under a stronger semantics. The aim is to find a set of ABox modifications (deletions and additions), called a repair plan, that addresses as many of the defects as possible. After formalizing this problem and introducing different notions of optimality, we investigate the computational complexity of reasoning about optimal repair plans and propose interactive algorithms for computing such plans. For deletion-only repair plans, we propose an improved algorithm and present the implementation of its core components in our CQAPri system. While we first focus on IAR and brave semantics, since the erroneous answers should ideally not hold under brave semantics whereas the desired answers should hold under IAR semantics, we investigate the use of the AR semantics in the fourth section. Indeed, the AR semantics is a natural alternative to IAR. We will see that even if considering the AR semantics changes the complexity of recognizing an optimal repair plan, we do not have to modify our algorithms to handle it. The main results of this chapter have been published in [Bienvenu *et al.* 2016b].

5.1 Query-driven repairing problem

While inconsistency-tolerant semantics are essential for returning useful results when consistency cannot be achieved, they by no means replace the need for tools for improving data quality. That is why we propose a complementary approach that exploits user feedback about query results to identify and correct errors.

There are several reasons to use queries to guide the repairing process. First, we note that it is typically impossible (for lack of time or information) to clean the entire dataset, and therefore reasonable to focus the effort on the parts of the data that are most relevant to users' needs. In the database arena, this observation has inspired work on integrating entity resolution into the querying process [Altwaijry *et al.* 2013]. Second, expert users may have a good idea of which answers are expected for queries concerning their area of expertise, and thus queries provide a natural way of identifying flaws. Indeed, it was recently proposed in [Kontokostas *et al.* 2014] to use queries to search for errors and help evaluate linked data quality. Finally, even non-expert users may notice anomalies when examining query results, and it would be a shame not to capitalize on this information, and in this way, help distribute the costly and time-consuming task of improving data quality, as argued in [Bergman *et al.* 2015].

We consider the following scenario: a user interacts with an OMQA system, posing conjunctive queries and receiving the results under inconsistency-tolerant semantics. When reviewing the results, the user detects some *unwanted answers*, which are erroneous and should therefore not have been retrieved, and identifies *wanted answers*, which should definitely be considered answers. Ideally, the unwanted tuples should not be returned as possible (brave) answers, and all of the desired tuples should be found among the sure (IAR) answers. To fix the detected problems and improve the quality of the data, the objective is to modify the ABox with a set of atomic changes (deletions and additions of facts), called a *repair plan*, that achieves as many of these objectives as possible, subject to the constraint that the changes must be validated by the user.

Our framework is inspired by that of [Jiménez-Ruiz *et al.* 2011], in which a user specifies two sets of axioms that should be entailed or not by a KB. Repair plans are introduced as pairs of sets of axioms to remove and add to obtain an ontology satisfying these requirements. Compared to prior work, distinguishing features of our framework are the specification of changes at the level of CQ answers, the use of inconsistency-tolerant semantics, and the introduction of optimality measures to handle situations in which not all objectives can be achieved.

Example 5.1.1. As a running example, we consider the simple KB $\mathcal{K}_{ex} = \langle \mathcal{T}_{ex}, \mathcal{A}_{ex} \rangle$.

$$\begin{aligned}\mathcal{T}_{ex} =& \{\mathsf{AProf} \sqsubseteq \mathsf{Prof}, \mathsf{FProf} \sqsubseteq \mathsf{Prof}, \exists \mathsf{Advise} \sqsubseteq \mathsf{Prof}, \mathsf{Prof} \sqsubseteq \mathsf{PhD}, \mathsf{Postdoc} \sqsubseteq \mathsf{PhD},\\ & \mathsf{AProf} \sqsubseteq \neg \mathsf{FProf}, \mathsf{Postdoc} \sqsubseteq \neg \mathsf{Prof}\}\\ \mathcal{A}_{ex} =& \{\mathsf{Postdoc}(a), \mathsf{AProf}(a), \mathsf{Advise}(a,b), \mathsf{Teach}(a,c)\}\end{aligned}$$

A user poses the queries $\mathsf{Prof}(x)$ and $\mathsf{PhD}(x)$ over $\mathcal{K}_{ex}$. Since $\mathcal{K}_{ex} \models_{\text{brave}} \mathsf{Prof}(a)$ and $\mathcal{K}_{ex} \models_{\text{AR}} \mathsf{PhD}(a)$, he receives a as an answer for both queries. Suppose that the user knows that a is definitely a PhD holder, but is not a professor. He would therefore like $\mathsf{PhD}(a)$ to hold under IAR semantics rather than AR which is less strong, and $\mathsf{Prof}(a)$ not to hold under any semantics, since any cause for this query has something wrong. He then indicates a as an unwanted answer for $\mathsf{Prof}(x)$ but a wanted answer for $\mathsf{PhD}(x)$. $\triangleleft$

A first way of repairing the data is to delete assertions from the ABox that lead to undesirable consequences, either because they contribute to the derivation of an unwanted answer or because they conflict with causes of some wanted answer.

Example 5.1.2 (Example 5.1.1 cont'd)**.** Deleting the assertions $\mathsf{AProf}(a)$ and $\mathsf{Advise}(a,b)$ from $\mathcal{A}_{ex}$ achieves the objectives since $\langle \mathcal{T}_{ex}, \{\mathsf{Postdoc}(a), \mathsf{Teach}(a,c)\}\rangle \not\models_{\text{brave}} \mathsf{Prof}(a)$ and $\langle \mathcal{T}_{ex}, \{\mathsf{Postdoc}(a), \mathsf{Teach}(a,c)\}\rangle \models_{\text{IAR}} \mathsf{PhD}(a)$. $\triangleleft$

The next example shows that, due to data incompleteness, it can also be necessary to add new assertions.

Example 5.1.3 (Example 5.1.1 cont'd)**.** Consider $\mathcal{K} = \langle \mathcal{T}_{ex}, \{\mathsf{AProf}(a)\}\rangle$ with the same wanted and unwanted answers as in Example 5.1.1. The assertion $\mathsf{AProf}(a)$ has to be removed to satisfy the unwanted answer, but then there remains no cause for the wanted answer. This is due to the fact that the only cause of $\mathsf{PhD}(a)$ in $\mathcal{K}$ contains an erroneous assertion: there is no "good" reason in the initial ABox for $\mathsf{PhD}(a)$ to hold. A solution is for the user to add a cause he knows for $\mathsf{PhD}(a)$, such as $\mathsf{Postdoc}(a)$. $\triangleleft$

We now provide a formal definition of the query-driven repairing problem investigated in this chapter.

Definition 5.1.4. A *query-driven repairing problem (QRP)* consists of a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ to repair and two sets $\mathcal{W}$,$\mathcal{U}$ of BCQs that $\mathcal{K}$ should entail ($\mathcal{W}$) or not entail ($\mathcal{U}$). A *repair plan (for $\mathcal{A}$)* is a pair $\mathcal{P} = (\mathcal{P}_-, \mathcal{P}_+)$ such that $\mathcal{P}_- \subseteq \mathcal{A}$ and $\mathcal{P}_+ \cap \mathcal{A} = \emptyset$; if $\mathcal{P}_+ = \emptyset$, we say that $\mathcal{P}$ is *deletion-only*.

The sets $\mathcal{U}$ and $\mathcal{W}$ correspond to the unwanted and wanted answers in our scenario: $q(\vec{a}) \in \mathcal{U}$ (resp. $\mathcal{W}$) means that $\vec{a}$ is an unwanted (resp. wanted) answer for q. Slightly abusing terminology, we will use the term *unwanted (resp. wanted) answers* to refer to the BCQs in $\mathcal{U}$ (resp. $\mathcal{W}$). We say that a repair plan $(\mathcal{P}_-, \mathcal{P}_+)$ *addresses all defects* of a QRP $(\mathcal{K}, \mathcal{W}, \mathcal{U})$ if the KB $\mathcal{K}' = \langle \mathcal{T}, (\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+\rangle$ is such that $\mathcal{K}' \models_{\text{IAR}} q$ for every $q \in \mathcal{W}$, and $\mathcal{K}' \not\models_{\text{brave}} q$ for every $q \in \mathcal{U}$.

The next example shows that by considering several answers at the same time, we can exploit the interaction between the different answers to reduce the search space.

Example 5.1.5 (Example 5.1.1 cont'd)**.** Consider the KB $\mathcal{K} = \langle \mathcal{T}_{ex}, \mathcal{A}\rangle$ with ABox $\mathcal{A} = \{\mathsf{Prof}(a), \mathsf{AProf}(b), \mathsf{FProf}(b), \mathsf{Teach}(a,c), \mathsf{Teach}(b,c), \mathsf{GradCourse}(c), \mathsf{TakeCourse}(s,c)\}$. It is easy to see that $\mathcal{K}$ is inconsistent, and its two repairs are obtained by removing either $\mathsf{AProf}(b)$ or $\mathsf{FProf}(b)$. Evaluating the queries $q_1(x) = \mathsf{PhD}(x)$ and $q_2(x) = \exists yz \mathsf{Prof}(x) \land \mathsf{Teach}(x,y) \land \mathsf{GradCourse}(y) \land \mathsf{TakeCourse}(z,y)$ over this KB yields:

$$\mathcal{K} \models_{\text{brave}} q_1(b) \quad \mathcal{K} \models_{\text{brave}} q_2(b) \quad \mathcal{K} \models_{\text{IAR}} q_2(a).$$

We consider the QRP $(\mathcal{K}, \mathcal{W}, \mathcal{U})$ with wanted answers $\mathcal{W} = \{q_1(b), q_2(a)\}$ and unwanted answers $\mathcal{U} = \{q_2(b)\}$.

Two deletion-only repair plans address all defects: $\{\mathsf{AProf}(b), \mathsf{Teach}(b,c)\}$ and $\{\mathsf{FProf}(b), \mathsf{Teach}(b,c)\}$. Indeed, we must delete exactly one of $\mathsf{AProf}(b)$ and $\mathsf{FProf}(b)$ for $q_1(b)$ to be entailed under IAR semantics, and we cannot remove $\mathsf{GradCourse}(c)$ or $\mathsf{TakeCourse}(s,c)$ without losing the wanted answer $q_2(a)$. Thus, the only way to get rid of $q_2(b)$ is to delete $\mathsf{Teach}(b,c)$.

If we consider only $\mathcal{U}$ (i.e. $\mathcal{W} = \emptyset$), there are additional possibilities such as $\{\mathsf{GradCourse}(c)\}$ and $\{\mathsf{TakeCourse}(s,c)\}$, and there is no evidence that $\mathsf{Teach}(b,c)$ has to be deleted. $\triangleleft$

If we want to avoid introducing new errors, a fully automated repairing process is impossible: we need the user to validate every assertion that is removed or added in order to remove (resp. add) only assertions that are false (resp. true).

Example 5.1.6 (Example 5.1.1 cont'd)**.** Reconsider the problem from Example 5.1.5, and suppose that the user knows that $\mathsf{FProf}(b)$ and $\mathsf{TakeCourse}(s,c)$ are false and every other assertion in $\mathcal{A}$ is true. An automatic repairing will remove the true assertion $\mathsf{Teach}(b,c)$. The problem is due to the absence of a "good" cause for the wanted answer $q_2(a)$ in $\mathcal{A}$. ◁

Since we will be studying an *interactive* repairing process, in which users must validate changes, we will also need to formalize the user's knowledge. For the purposes of this work, we assume that the user's knowledge is consistent with the considered TBox $\mathcal{T}$, and so can be captured as a set $\mathcal{M}_{\mathsf{user}}$ of models of $\mathcal{T}$. Instead of using $\mathcal{M}_{\mathsf{user}}$ directly, it will be more convenient to work with the *function* user *induced from* $\mathcal{M}_{\mathsf{user}}$ that assigns truth values to BCQs in the obvious way: $\mathsf{user}(q) = \mathsf{true}$ if q is true in every $\mathcal{I} \in \mathcal{M}_{\mathsf{user}}$, $\mathsf{user}(q) = \mathsf{false}$ if q is false in every $\mathcal{I} \in \mathcal{M}_{\mathsf{user}}$, and $\mathsf{user}(q) = \mathsf{unknown}$ otherwise. We will assume throughout the chapter the following *truthfulness condition*: $\mathsf{user}(q) = \mathsf{false}$ for every $q \in \mathcal{U}$, and $\mathsf{user}(q) = \mathsf{true}$ for every $q \in \mathcal{W}$.

We now formalize the requirement that repair plans only contain changes that are sanctioned by the user.

Definition 5.1.7. A repair plan $(\mathcal{P}_-, \mathcal{P}_+)$ is *validatable w.r.t.* user[1] just in the case that $\mathsf{user}(\alpha) = \mathsf{false}$ for every $\alpha \in \mathcal{P}_-$ and $\mathsf{user}(\alpha) = \mathsf{true}$ for every $\alpha \in \mathcal{P}_+$.

Unfortunately, it may be the case that there is no validatable repair plan addressing all defects. This may happen, for instance, if the user knows some answer is wrong but cannot pinpoint which assertion is at fault, as we illustrate next.

Example 5.1.8 (Example 5.1.1 cont'd)**.** Consider the QRP given by:

$$\begin{aligned}
\mathcal{K} =& \langle \mathcal{T}_{ex}, \{\mathsf{FProf}(a), \mathsf{Teach}(a,c), \mathsf{GradCourse}(c)\}\rangle \\
\mathcal{W} =& \{\mathsf{Prof}(a)\} \\
\mathcal{U} =& \{\exists x \mathsf{Prof}(a) \wedge \mathsf{Teach}(a,x) \wedge \mathsf{GradCourse}(x)\}
\end{aligned}$$

Suppose that $\mathsf{user}(\mathsf{FProf}(a)) = \mathsf{false}$, $\mathsf{user}(\mathsf{Teach}(a,c)) = \mathsf{unknown}$, $\mathsf{user}(\mathsf{GradCourse}(c)) = \mathsf{unknown}$, and $\mathsf{user}(\mathsf{AProf}(a)) = \mathsf{true}$. It is not possible to satisfy the wanted and unwanted answers at the same time, since adding the true assertion $\mathsf{AProf}(a)$ creates a cause for the unwanted answer that does not contains any assertion α with $\mathsf{user}(\alpha) = \mathsf{false}$: the user does not know which of $\mathsf{Teach}(a,c)$ and $\mathsf{GradCourse}(c)$ is erroneous. ◁

As validatable repair plans addressing all defects are not guaranteed to exist, our aim will be to find repair plans that are optimal in the sense that they address as many of the defects as possible, subject to the constraint that the changes must be validated by the user.

5.2 Optimal repair plans

To compare repair plans, we consider the answers from $\mathcal{U}$ and $\mathcal{W}$ that are satisfied by the resulting KBs, where:

[1] In what follows, we often omit "w.r.t. user" and leave it implicit.

- $q \in \mathcal{U}$ is *satisfied by* $\mathcal{K}$ if $\mathcal{K} \not\models_{\text{brave}} q$;
- $q \in \mathcal{W}$ is *satisfied by* $\mathcal{K}$ if there exists $\mathcal{C} \in \mathsf{causes}(q, \mathcal{K})$ such that $\mathsf{confl}(\mathcal{C}, \mathcal{K}) = \emptyset$ and there is no $\alpha \in \mathcal{C}$ with $\mathsf{user}(\alpha) = \mathsf{false}$.

Remark 5.2.1. Observe that for $q \in \mathcal{W}$ to be satisfied by $\mathcal{K}$, we require not only that $\mathcal{K} \models_{\text{IAR}} q$, but also that there exists a cause for q that does not contain any assertions known to be false, i.e. $\mathcal{K} \models_{\text{IAR}} q$ should hold "for a good reason". We impose this additional requirement to avoid counterintuitive situations, e.g. preferring repair plans that remove fewer false assertions in order to retain a conflict-free (but erroneous) cause for a wanted answer.

We say that a repair plan $\mathcal{P} = (\mathcal{P}_-, \mathcal{P}_+)$ *satisfies* $q \in \mathcal{U} \cup \mathcal{W}$ if the KB $\mathcal{K}_\mathcal{P} = \langle \mathcal{T}, (\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+ \rangle$ satisfies q, and we use $\mathcal{S}(\mathcal{P})$ (resp. $\mathcal{S}_\mathcal{U}(\mathcal{P})$, $\mathcal{S}_\mathcal{W}(\mathcal{P})$) to denote the sets of answers (resp. unwanted answers, wanted answers) satisfied by $\mathcal{P}$.

Two repair plans $\mathcal{P}$ and $\mathcal{P}'$ can be compared w.r.t. the sets of unwanted and wanted answers that they satisfy: for $A \in \{\mathcal{U}, \mathcal{W}\}$, we define the preorder $\preceq_A$ by setting $\mathcal{P} \preceq_A \mathcal{P}'$ iff $\mathcal{S}_A(\mathcal{P}) \subseteq \mathcal{S}_A(\mathcal{P}')$, and obtain the corresponding strict order ($\prec_A$) and equivalence relations ($\sim_A$) in the usual way. If the two criteria are equally important, we can combine them using the Pareto principle:

$$\mathcal{P} \preceq_{\{\mathcal{U},\mathcal{W}\}} \mathcal{P}' \text{ iff } \mathcal{P} \preceq_\mathcal{U} \mathcal{P}' \text{ and } \mathcal{P} \preceq_\mathcal{W} \mathcal{P}'.$$

Alternatively, we can use the lexicographic method to give priority either to the wanted answers ($\preceq_{\mathcal{W},\mathcal{U}}$) or unwanted answers ($\preceq_{\mathcal{U},\mathcal{W}}$):

$$\mathcal{P} \preceq_{A,B} \mathcal{P}' \text{ iff } \mathcal{P} \prec_A \mathcal{P}' \text{ or } \mathcal{P} \sim_A \mathcal{P}' \text{ and } \mathcal{P} \preceq_B \mathcal{P}', \text{ where } \{A, B\} = \{\mathcal{U}, \mathcal{W}\}.$$

For each of the preceding preference relations $\preceq$, we can define the corresponding notions of $\preceq$-optimal repair plan.

Definition 5.2.2 (Optimal repair plan)**.** A repair plan $(\mathcal{P}_-, \mathcal{P}_+)$ is *globally (resp. locally) $\preceq$-optimal w.r.t.* user iff it is validatable w.r.t. user and there is no other validatable repair plan $(\mathcal{P}'_-, \mathcal{P}'_+)$ such that $(\mathcal{P}_-, \mathcal{P}_+) \prec (\mathcal{P}'_-, \mathcal{P}'_+)$ (resp. such that $\mathcal{P}_- \subseteq \mathcal{P}'_-$, $\mathcal{P}_+ \subseteq \mathcal{P}'_+$ and $(\mathcal{P}_-, \mathcal{P}_+) \prec (\mathcal{P}'_-, \mathcal{P}'_+)$).

Globally $\preceq$-optimal repair plans are those that are maximal with respect to the preference relation $\preceq$, whereas locally $\preceq$-optimal repair plans are those that cannot be improved in the $\preceq$ ordering by adding further assertions to $\mathcal{P}_-$ or $\mathcal{P}_+$.

Remark 5.2.3. If a repair plan is validatable and addresses all defects of a QRP, then it is globally $\preceq_\mathcal{U}$-optimal. If it additionally satisfies every $q \in \mathcal{W}$ (ensuring that there is a "good" cause for every $q \in \mathcal{W}$), then it is globally $\preceq$-optimal for every $\preceq \in \{\preceq_\mathcal{W}, \preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$.

The following example illustrates the difference between local and global optimality.

Example 5.2.4. Consider the QRP $(\langle \mathcal{T}_{ex}, \mathcal{A}\rangle, \mathcal{W}, \mathcal{U})$ where

$$\begin{aligned}
\mathcal{A} =& \{\mathsf{Teach}(a,e), \mathsf{Advise}(a,b), \mathsf{TakeCourse}(b,c), \mathsf{TakeCourse}(b,e), \mathsf{GradCourse}(e)\}\\
\mathcal{W} =& \{\exists x \mathsf{Teach}(a,x), \exists x \mathsf{TakeCourse}(b,x) \wedge \mathsf{GradCourse}(x)\}\\
\mathcal{U} =& \{\exists xy \mathsf{Teach}(a,x) \wedge \mathsf{Advise}(a,y) \wedge \mathsf{TakeCourse}(y,x) \wedge \mathsf{GradCourse}(x)\}
\end{aligned}$$

Suppose that $\mathsf{user}(\mathsf{Teach}(a,e)) = \mathsf{user}(\mathsf{GradCourse}(e)) = \mathsf{false}$, $\mathsf{user}(\alpha) = \mathsf{unknown}$ for the other $\alpha \in \mathcal{A}$, and the user knows that $\mathsf{Teach}(a,c)$, $\mathsf{Teach}(a,d)$ and $\mathsf{GradCourse}(c)$ are true.

It can be verified that the repair plan $\mathcal{P}_1 = (\{\mathsf{Teach}(a,e), \mathsf{GradCourse}(e)\}, \{\mathsf{Teach}(a,c)\})$ satisfies the first answer in $\mathcal{W}$ and the (only) answer in $\mathcal{U}$. It is locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal since the only way to satisfy the second wanted answer would be to add $\mathsf{GradCourse}(c)$, which would create a cause for the unwanted answer, which could not be repaired by removing additional assertions as the user does not know which of $\mathsf{Advise}(a,b)$ and $\mathsf{TakeCourse}(b,c)$ is false. However, $\mathcal{P}_1$ is not globally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal because $\mathcal{P}_2 = (\{\mathsf{Teach}(a,e), \mathsf{GradCourse}(e)\}, \{\mathsf{Teach}(a,d), \mathsf{GradCourse}(c)\})$ satisfies all answers in $\mathcal{W} \cup \mathcal{U}$. $\triangleleft$

5.2.1 Characterization and complexity analysis

In order to gain a better understanding of the computational properties of the different ways of ranking repair plans, we study the complexity of deciding if a given repair plan is optimal w.r.t. the different criteria. Since validatability of a repair plan depends on user, in this section, we will use the following notations for the sets of false, unknown, and true ABox assertions w.r.t. user:

$$\begin{aligned}
False_{\mathsf{user}} &= \{\alpha \in \mathcal{A} \mid \mathsf{user}(\alpha) = \mathsf{false}\}\\
Unk_{\mathsf{user}} &= \{\alpha \in \mathcal{A} \mid \mathsf{user}(\alpha) = \mathsf{unknown}\}\\
True_{\mathsf{user}} &= \{\alpha \mid \mathsf{user}(\alpha) = \mathsf{true}\}
\end{aligned}$$

Checking if an assertion is false (resp. unknown, true) is in P w.r.t. the size of $False_{\mathsf{user}}$ (resp. Unk_{user}, $True_{\mathsf{user}}$). The sets $False_{\mathsf{user}}$ and Unk_{user} are included in $\mathcal{A}$, while $True_{\mathsf{user}}$ may be larger. However, only the assertions of $True_{\mathsf{user}}$ that are relevant to the given QRP need to be considered. We make similar assumptions as for data complexity and thus assume that the sizes of the queries and the TBox are bounded. We thus measure complexity w.r.t. $|\mathcal{A}|$, $|\mathcal{U}|$, $|\mathcal{W}|$, as well as the size of the set

$$\begin{aligned}
True^{rel}_{\mathsf{user}} = \{\alpha \in True_{\mathsf{user}} \mid & \text{ there exists } q \in \mathcal{W} \text{ such that}\\
& \alpha \in \mathcal{C} \text{ for some } \mathcal{C} \in \mathsf{causes}(q, \mathcal{A} \cup True_{\mathsf{user}})\}
\end{aligned}$$

We make the reasonable assumption that $True_{\mathsf{user}}$ (hence $True^{rel}_{\mathsf{user}}$) is finite.

We begin with the following lemma which shows that removing false assertions or adding true assertions (whose conflicts are false) can only satisfy more wanted answers, and removing additional false assertions, while adding the same set of true assertions, can only satisfy more unwanted answers.

Lemma 5.2.5. *Let* $(\mathcal{P}_-, \mathcal{P}_+)$ *and* $(\mathcal{P}'_-, \mathcal{P}'_+)$ *be validatable repair plans.*

1. *If* $\mathcal{P}_- \subseteq \mathcal{P}'_-$ *and* $\mathcal{P}_+ \subseteq \mathcal{P}'_+$, *then* $\mathcal{S}_{\mathcal{W}}(\mathcal{P}_-, \mathcal{P}_+) \subseteq \mathcal{S}_{\mathcal{W}}(\mathcal{P}'_-, \mathcal{P}'_+)$.
2. *If* $\mathcal{P}_- \subseteq \mathcal{P}'_-$ *and* $\mathcal{P}_+ = \mathcal{P}'_+$, *then* $\mathcal{S}_{\mathcal{U}}(\mathcal{P}_-, \mathcal{P}_+) \subseteq \mathcal{S}_{\mathcal{U}}(\mathcal{P}'_-, \mathcal{P}'_+)$.

Proof. Suppose that $\mathcal{P}_- \subseteq \mathcal{P}'_-$ and $\mathcal{P}_+ \subseteq \mathcal{P}'_+$ and let $q \in \mathcal{S}_{\mathcal{W}}(\mathcal{P}_-, \mathcal{P}_+)$. There exists a cause $\mathcal{C}$ for q in $(\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+$ such that $\mathcal{C}$ does not contain any false assertion and has no conflicts in $(\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+$. Since $\mathcal{C} \subseteq \mathcal{A} \cup \mathcal{P}_+$ and $\mathcal{P}_+ \subseteq \mathcal{P}'_+$, $\mathcal{C} \subseteq \mathcal{A} \cup \mathcal{P}'_+$, and since $\mathcal{C}$ does not contain any false assertion and $(\mathcal{P}'_-, \mathcal{P}'_+)$ is validatable, $\mathcal{C} \cap \mathcal{P}'_- = \emptyset$, so $\mathcal{C} \subseteq (\mathcal{A} \backslash \mathcal{P}'_-) \cup \mathcal{P}'_+$. Moreover, $\mathcal{C}$ has no conflict in $(\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+$, so the set of assertions of $\mathcal{A}$ in conflict with $\mathcal{C}$ is included in $\mathcal{P}_- \subseteq \mathcal{P}'_-$, so $\mathcal{C}$ has no conflict in $(\mathcal{A} \backslash \mathcal{P}'_-) \cup \mathcal{P}'_+$ (note that since the assertions of $\mathcal{C}$ are nonfalse, and the repair plans are validatable, assertions of $\mathcal{C}$ cannot conflict with assertions of $\mathcal{P}'_+$). It follows that $q \in \mathcal{S}_{\mathcal{W}}(\mathcal{P}'_-, \mathcal{P}'_+)$.

Suppose that $\mathcal{P}_- \subseteq \mathcal{P}'_-$ and $\mathcal{P}_+ = \mathcal{P}'_+$ and let $q \in \mathcal{S}_{\mathcal{U}}(\mathcal{P}_-, \mathcal{P}_+)$. There is no cause for q in $(\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+ \subseteq (\mathcal{A} \backslash \mathcal{P}'_-) \cup \mathcal{P}'_+$, so $q \in \mathcal{S}_{\mathcal{U}}(\mathcal{P}'_-, \mathcal{P}'_+)$. $\square$

We next provide characterizations of optimal plans in terms of the notion of *satisfiability* of answers.

Definition 5.2.6 (Satisfiable answer)**.** An answer $q \in \mathcal{U} \cup \mathcal{W}$ is *satisfiable* if there exists a validatable repair plan that satisfies q. We say that q is *satisfiable w.r.t. a validatable repair plan* $\mathcal{P} = (\mathcal{P}_-, \mathcal{P}_+)$ if there exists a validatable repair plan $\mathcal{P}' = (\mathcal{P}'_-, \mathcal{P}'_+)$ such that $\mathcal{P}_- \subseteq \mathcal{P}'_-$, $\mathcal{P}_+ \subseteq \mathcal{P}'_+$, $q \in \mathcal{S}(\mathcal{P}')$, and $\mathcal{P} \preceq_{\{\mathcal{U},\mathcal{W}\}} \mathcal{P}'$.

Proposition 5.2.7. *A validatable repair plan* $\mathcal{P}$ *is:*

- *globally* $\preceq_{\mathcal{U}}$- *(resp.* $\preceq_{\mathcal{W}}$-*) optimal iff it is locally* $\preceq_{\mathcal{U}}$- *(resp.* $\preceq_{\mathcal{W}}$-*) optimal iff it satisfies every satisfiable* $q \in \mathcal{U}$ *(resp.* $q \in \mathcal{W}$*).*
- *locally* $\preceq_{\mathcal{U},\mathcal{W}}$*-optimal iff it is locally* $\preceq_{\{\mathcal{U},\mathcal{W}\}}$*-optimal iff it satisfies every* $q \in \mathcal{U} \cup \mathcal{W}$ *that is satisfiable w.r.t.* $\mathcal{P}$.
- *locally* $\preceq_{\mathcal{W},\mathcal{U}}$*-optimal iff it satisfies every satisfiable* $q \in \mathcal{W}$ *and every* $q \in \mathcal{U}$ *that is satisfiable w.r.t.* $\mathcal{P}$.

Proof.
• A validatable repair plan is globally (or locally) $\preceq_{\mathcal{U}}$- (resp. $\preceq_{\mathcal{W}}$-) optimal iff satisfies every satisfiable $q \in \mathcal{U}$ (resp. $q \in \mathcal{W}$):
- Let $(\mathcal{P}_-, \mathcal{P}_+)$ be a globally (or locally) $\preceq_{\mathcal{U}}$- (resp. $\preceq_{\mathcal{W}}$-) optimal repair plan. Take some satisfiable $q \in \mathcal{U}$ (resp. $q \in \mathcal{W}$), and let $(\mathcal{P}'_-, \mathcal{P}'_+)$ be a validatable repair plan satisfying q. By Lemma 5.2.5, $(\mathcal{P}_-, \mathcal{P}_+) \preceq_{\mathcal{U}} (\mathcal{P}_- \cup \mathcal{P}'_-, \mathcal{P}_+)$ (resp. $(\mathcal{P}_-, \mathcal{P}_+) \preceq_{\mathcal{W}} (\mathcal{P}_- \cup \mathcal{P}'_-, \mathcal{P}_+ \cup \mathcal{P}'_+)$). Because of global (or local) optimality, we must in fact have $(\mathcal{P}_-, \mathcal{P}_+) \sim_{\mathcal{U}} (\mathcal{P}_- \cup \mathcal{P}'_-, \mathcal{P}_+)$ (resp. $(\mathcal{P}_-, \mathcal{P}_+) \sim_{\mathcal{W}} (\mathcal{P}_- \cup \mathcal{P}'_-, \mathcal{P}_+ \cup \mathcal{P}'_+)$), and so q is satisfied by $(\mathcal{P}_-, \mathcal{P}_+)$.

- In the other direction, it follows from the definition of satisfiable answers that if a validatable repair plan satisfies every satisfiable $q \in \mathcal{U}$ (resp. $q \in \mathcal{W}$), it is globally (so also locally) $\preceq_{\mathcal{U}}$- (resp. $\preceq_{\mathcal{W}}$-) optimal.

• A validatable repair plan is locally $\preceq_{\mathcal{U},\mathcal{W}}$-optimal iff it is locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal:
- If a repair plan $(\mathcal{P}_-,\mathcal{P}_+)$ is locally $\preceq_{\mathcal{U},\mathcal{W}}$-optimal, it is locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal, otherwise there would be a validatable repair plan $(\mathcal{P}'_-,\mathcal{P}'_+)$ such that $\mathcal{P}_- \subseteq \mathcal{P}'_-$, $\mathcal{P}_+ \subseteq \mathcal{P}'_+$ and $(\mathcal{P}_-,\mathcal{P}_+) \prec_{\{\mathcal{U},\mathcal{W}\}} (\mathcal{P}'_-,\mathcal{P}'_+)$, so also such that $(\mathcal{P}_-,\mathcal{P}_+) \prec_{\mathcal{U},\mathcal{W}} (\mathcal{P}'_-,\mathcal{P}'_+)$.
- Suppose for a contradiction that a repair plan $(\mathcal{P}_-,\mathcal{P}_+)$ is locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal and not locally $\preceq_{\mathcal{U},\mathcal{W}}$-optimal. Then there exists a validatable repair plan $(\mathcal{P}'_-,\mathcal{P}'_+)$ such that $\mathcal{P}_- \subseteq \mathcal{P}'_-$, $\mathcal{P}_+ \subseteq \mathcal{P}'_+$ and $(\mathcal{P}_-,\mathcal{P}_+) \prec_{\mathcal{U},\mathcal{W}} (\mathcal{P}'_-,\mathcal{P}'_+)$. Since removing more false assertions cannot deteriorate satisfied wanted answers (see Lemma 5.2.5), $(\mathcal{P}'_-,\mathcal{P}'_+)$ cannot satisfy more unwanted answers otherwise we would have $(\mathcal{P}_-,\mathcal{P}_+) \prec_{\{\mathcal{U},\mathcal{W}\}} (\mathcal{P}_- \cup \mathcal{P}'_-,\mathcal{P}_+)$. Hence $(\mathcal{P}'_-,\mathcal{P}'_+)$ must satisfy the same unwanted answers and more wanted answers, which yields $(\mathcal{P}_-,\mathcal{P}_+) \prec_{\{\mathcal{U},\mathcal{W}\}} (\mathcal{P}'_-,\mathcal{P}'_+)$, contradicting our assumption of local $\preceq_{\mathcal{U},\mathcal{W}}$-optimality.

• A validatable repair plan $\mathcal{P}$ is locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$- ($\preceq_{\mathcal{U},\mathcal{W}}$-) optimal iff it satisfies every $q \in \mathcal{U} \cup \mathcal{W}$ that is satisfiable w.r.t. $\mathcal{P}$:
- Suppose that $(\mathcal{P}_-,\mathcal{P}_+)$ is locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal, and let $q \in \mathcal{U} \cup \mathcal{W}$ be an answer that is satisfiable w.r.t. $(\mathcal{P}_-,\mathcal{P}_+)$. Then there exists a validatable repair plan $(\mathcal{P}'_-,\mathcal{P}'_+)$ such that $\mathcal{P}_- \subseteq \mathcal{P}'_-$, $\mathcal{P}_+ \subseteq \mathcal{P}'_+$ and $(\mathcal{P}_-,\mathcal{P}_+) \preceq_{\{\mathcal{U},\mathcal{W}\}} (\mathcal{P}'_-,\mathcal{P}'_+)$ and $q \in \mathcal{S}(\mathcal{P}'_-,\mathcal{P}'_+)$. Since $(\mathcal{P}_-,\mathcal{P}_+)$ is locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal, we must have $(\mathcal{P}_-,\mathcal{P}_+) \sim_{\{\mathcal{U},\mathcal{W}\}} (\mathcal{P}'_-,\mathcal{P}'_+)$, and hence $q \in \mathcal{S}(\mathcal{P}_-,\mathcal{P}_+)$.
- In the other direction, suppose that $(\mathcal{P}_-,\mathcal{P}_+)$ is a validatable repair plan that satisfies every $q \in \mathcal{U} \cup \mathcal{W}$ that is satisfiable w.r.t. $(\mathcal{P}_-,\mathcal{P}_+)$. Consider a validatable repair plan $(\mathcal{P}'_-,\mathcal{P}'_+)$ such that $\mathcal{P}_- \subseteq \mathcal{P}'_-$, $\mathcal{P}_+ \subseteq \mathcal{P}'_+$ and $(\mathcal{P}_-,\mathcal{P}_+) \preceq_{\{\mathcal{U},\mathcal{W}\}} (\mathcal{P}'_-,\mathcal{P}'_+)$, and take some $q \in \mathcal{S}(\mathcal{P}'_-,\mathcal{P}'_+)$. Then q is satisfiable w.r.t. $(\mathcal{P}_-,\mathcal{P}_+)$, so, by our assumption, it must be satisfied by $(\mathcal{P}_-,\mathcal{P}_+)$. We thus have $(\mathcal{P}_-,\mathcal{P}_+) \sim_{\{\mathcal{U},\mathcal{W}\}} (\mathcal{P}'_-,\mathcal{P}'_+)$, so $(\mathcal{P}_-,\mathcal{P}_+)$ is locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal.

• A validatable repair plan $\mathcal{P}$ is locally $\preceq_{\mathcal{W},\mathcal{U}}$-optimal iff it satisfies every satisfiable $q \in \mathcal{W}$ and every $q \in \mathcal{U}$ that is satisfiable w.r.t. $\mathcal{P}$:
- Suppose that $(\mathcal{P}_-,\mathcal{P}_+)$ is locally $\preceq_{\mathcal{W},\mathcal{U}}$-optimal. First consider some satisfiable $q \in \mathcal{W}$. Then there exists a validatable repair plan $(\mathcal{P}'_-,\mathcal{P}'_+)$ such that $q \in \mathcal{S}(\mathcal{P}'_-,\mathcal{P}'_+)$. By Lemma 5.2.5, we have $(\mathcal{P}_-,\mathcal{P}_+) \preceq_{\mathcal{W}} (\mathcal{P}_- \cup \mathcal{P}'_-,\mathcal{P}_+ \cup \mathcal{P}'_+)$. Applying our assumption of local $\preceq_{\mathcal{W},\mathcal{U}}$-optimality, we have $(\mathcal{P}_-,\mathcal{P}_+) \sim_{\mathcal{W}} (\mathcal{P}_- \cup \mathcal{P}'_-,\mathcal{P}_+ \cup \mathcal{P}'_+)$, which implies that q is satisfied by $(\mathcal{P}_-,\mathcal{P}_+)$.

Next take some $q \in \mathcal{U}$ that is satisfiable w.r.t. $(\mathcal{P}_-,\mathcal{P}_+)$. Then there exists a validatable repair plan $(\mathcal{P}'_-,\mathcal{P}'_+)$ such that $\mathcal{P}_- \subseteq \mathcal{P}'_-$, $\mathcal{P}_+ \subseteq \mathcal{P}'_+$, $(\mathcal{P}_-,\mathcal{P}_+) \preceq_{\{\mathcal{U},\mathcal{W}\}} (\mathcal{P}'_-,\mathcal{P}'_+)$ and $q \in \mathcal{S}(\mathcal{P}'_-,\mathcal{P}'_+)$. Since $(\mathcal{P}_-,\mathcal{P}_+)$ is locally $\preceq_{\mathcal{W},\mathcal{U}}$-optimal, we must have $(\mathcal{P}_-,\mathcal{P}_+) \sim_{\mathcal{W}} (\mathcal{P}'_-,\mathcal{P}'_+)$ and $(\mathcal{P}_-,\mathcal{P}_+) \sim_{\mathcal{U}} (\mathcal{P}'_-,\mathcal{P}'_+)$. From the latter, we obtain $q \in \mathcal{S}(\mathcal{P}_-,\mathcal{P}_+)$.
- In the other direction, let $(\mathcal{P}_-,\mathcal{P}_+)$ be a validatable repair plan that satisfies every satisfiable $q \in \mathcal{W}$ and every $q \in \mathcal{U}$ that is satisfiable w.r.t. $(\mathcal{P}_-,\mathcal{P}_+)$. Take some validatable repair plan $(\mathcal{P}'_-,\mathcal{P}'_+)$ such that $\mathcal{P}_- \subseteq \mathcal{P}'_-$, $\mathcal{P}_+ \subseteq \mathcal{P}'_+$ and $(\mathcal{P}_-,\mathcal{P}_+) \preceq_{\mathcal{W},\mathcal{U}} (\mathcal{P}'_-,\mathcal{P}'_+)$. We observe that $(\mathcal{P}'_-,\mathcal{P}'_+)$ cannot satisfy more wanted answers than $(\mathcal{P}_-,\mathcal{P}_+)$ since $(\mathcal{P}_-,\mathcal{P}_+)$ satisfies all satisfiable wanted answers, nor can it satisfy more unwanted answers, since otherwise $(\mathcal{P}_-,\mathcal{P}_+)$ would not satisfy all unwanted answers that are satisfiable w.r.t. $(\mathcal{P}_-,\mathcal{P}_+)$. $\square$

The next lemma characterizes when a validatable repair plan satisfies an unwanted answer.

Lemma 5.2.8. *Let $(\mathcal{P}_-,\mathcal{P}_+)$ be a validatable repair plan. Then $(\mathcal{P}_-,\mathcal{P}_+)$ satisfies $q \in \mathcal{U}$ iff $\mathcal{P}_- \cap \mathcal{C} \neq \emptyset$ for every $\mathcal{C} \in \mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\cup\mathcal{P}_+\rangle)$.*

Proof. For the first direction, suppose that $(\mathcal{P}_-,\mathcal{P}_+)$ satisfies $q \in \mathcal{U}$. This means that $\langle\mathcal{T},(\mathcal{A}\backslash\mathcal{P}_-)\cup\mathcal{P}_+\rangle \not\models_{\text{brave}} q$. It follows that for every $\mathcal{C} \in \mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\cup\mathcal{P}_+\rangle)$, we have $\mathcal{C} \not\subseteq (\mathcal{A}\backslash\mathcal{P}_-)\cup\mathcal{P}_+$, hence $\mathcal{C}\cap\mathcal{P}_- \neq \emptyset$.

For the second direction, suppose that $\mathcal{P}_- \cap \mathcal{C} \neq \emptyset$ for every $\mathcal{C} \in \mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\cup\mathcal{P}_+\rangle)$. It follows that $\mathsf{causes}(q,\langle\mathcal{T},(\mathcal{A}\cup\mathcal{P}_+)\setminus\mathcal{P}_-\rangle) = \emptyset$. Since $(\mathcal{P}_-,\mathcal{P}_+)$ is validatable, we know that $\mathsf{user}(\alpha) = \mathsf{false}$ for every $\alpha \in \mathcal{P}_-$ and $\mathsf{user}(\alpha) = \mathsf{true}$ for every $\alpha \in \mathcal{P}_+$. In particular, this means that $\mathcal{P}_- \cap \mathcal{P}_+ = \emptyset$, so $(\mathcal{A}\cup\mathcal{P}_+)\setminus\mathcal{P}_- = (\mathcal{A}\setminus\mathcal{P}_-)\cup\mathcal{P}_+$. We therefore have $\mathsf{causes}(q,\langle\mathcal{T},(\mathcal{A}\setminus\mathcal{P}_-)\cup\mathcal{P}_+\rangle) = \emptyset$, hence $\langle\mathcal{T},(\mathcal{A}\backslash\mathcal{P}_-)\cup\mathcal{P}_+\rangle \not\models_{\text{brave}} q$. □

We now establish the following characterizations of satisfiable answers and answers satisfiable w.r.t. a repair plan.

Lemma 5.2.9. *An answer $q \in \mathcal{U}$ is satisfiable iff for every $\mathcal{C} \in \mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\rangle)$ there exists $\alpha \in \mathcal{C}$ such that $\mathsf{user}(\alpha) = \mathsf{false}$.*

Proof. If $q \in \mathcal{U}$ is satisfiable, then there exists a validatable repair plan $(\mathcal{P}_-,\mathcal{P}_+)$ that satisfies q. By Lemma 5.2.8, we must have $\mathcal{P}_-\cap\mathcal{C} \neq \emptyset$ for every $\mathcal{C} \in \mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\cup\mathcal{P}_+\rangle)$, hence for every $\mathcal{C} \in \mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\rangle)$. Since $(\mathcal{P}_-,\mathcal{P}_+)$ is validatable, we know that $\mathcal{P}_- \subseteq False_{\mathsf{user}}$, hence every cause of q in $\langle\mathcal{T},\mathcal{A}\rangle$ contains at least one assertion α such that $\mathsf{user}(\alpha) = \mathsf{false}$.

In the other direction, if for every $\mathcal{C} \in \mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\rangle)$ there exists $\alpha \in \mathcal{C}$ such that $\mathsf{user}(\alpha) = \mathsf{false}$, then it is easily shown using Lemma 5.2.8 that

$$(\{\alpha \mid \exists\mathcal{C} \in \mathsf{causes}(q,\langle\mathcal{T},\mathcal{A}\rangle), \alpha \in \mathcal{C}, \mathsf{user}(\alpha) = \mathsf{false}\},\emptyset)$$

is a validatable repair plan that satisfies q. □

Lemma 5.2.10. *An answer $q \in \mathcal{W}$ is satisfiable iff there exists a $\mathcal{T}$-consistent set of assertions $\mathcal{C}_0$ such that $\langle\mathcal{T},\mathcal{C}_0\rangle \models q$ and for every $\alpha \in \mathcal{C}_0$, either*

- $\mathsf{user}(\alpha) = \mathsf{true}$, *or*
- $\alpha \in \mathcal{A}$, $\mathsf{user}(\alpha) = \mathsf{unknown}$ *and for every $\beta \in \mathcal{A}$ such that $\langle\mathcal{T},\{\alpha,\beta\}\rangle \models \bot$*, $\mathsf{user}(\beta) = \mathsf{false}$.

(We will call $\mathcal{C}_0$ a witness *for the satisfiability of q.)*

Proof. If $q \in \mathcal{W}$ is satisfiable, then there exists a validatable repair plan $(\mathcal{P}_-,\mathcal{P}_+)$ such that $(\mathcal{A}\backslash\mathcal{P}_-)\cup\mathcal{P}_+$ contains a cause $\mathcal{C}_0$ for q that contains no false assertion and has no conflicts in $(\mathcal{A}\backslash\mathcal{P}_-)\cup\mathcal{P}_+$. It follows that for every $\alpha \in \mathcal{C}_0$, either $\alpha \in \mathcal{P}_+$ and $\mathsf{user}(\alpha) = \mathsf{true}$, or $\alpha \in \mathcal{A}$ and $\mathsf{user}(\alpha) = \mathsf{true}$ or $\mathsf{user}(\alpha) = \mathsf{unknown}$, and every conflict β of α is in $\mathcal{P}_-$, hence is such that $\mathsf{user}(\beta) = \mathsf{false}$.

In the other direction, if q and $\mathcal{C}_0$ satisfy the conditions of the lemma statement, then one can easily verify that

$$(\{\beta \in \mathcal{A} \mid \exists \alpha \in \mathcal{C}_0, \langle \mathcal{T}, \{\alpha, \beta\}\rangle \models \bot, \mathsf{user}(\beta) = \mathsf{false}\},\\ \{\alpha \in \mathcal{C}_0 \setminus \mathcal{A} \mid \mathsf{user}(\alpha) = \mathsf{true}\})$$

is a validatable repair plan that satisfies q. $\square$

Lemma 5.2.11. *Let $(\mathcal{P}_-, \mathcal{P}_+)$ be a validatable repair plan for the KB $\langle \mathcal{T}, \mathcal{A}\rangle$. Then an answer $q \in \mathcal{U}$ is satisfiable w.r.t. $(\mathcal{P}_-, \mathcal{P}_+)$ iff $q \in \mathcal{U}$ is satisfiable for the KB $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+\rangle$.*

Proof. If $q \in \mathcal{U}$ is satisfiable w.r.t. $(\mathcal{P}_-, \mathcal{P}_+)$, then there exists a validatable repair plan $(\mathcal{P}'_-, \mathcal{P}'_+)$ with $\mathcal{P}_- \subseteq \mathcal{P}'_-$ and $\mathcal{P}_+ \subseteq \mathcal{P}'_+$ that satisfies q. By Lemma 5.2.8, $\mathcal{P}'_-$ must intersect all of the causes of q w.r.t. $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}'_+\rangle$. Since $\mathcal{P}_+ \subseteq \mathcal{P}'_+$, the set $\mathcal{P}'_-$ intersects all of q's causes w.r.t. $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+\rangle$. By applying Lemma 5.2.8 again, we can show that the repair plan $(\mathcal{P}'_-, \emptyset)$ witnesses the satisfiability of q for the KB $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+\rangle$.

In the other direction, suppose that $q \in \mathcal{U}$ is satisfiable when $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+\rangle$ is the input KB. By Lemma 5.2.9, we know that for every $\mathcal{C} \in \mathsf{causes}(q, \langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+\rangle)$ there exists $\alpha \in \mathcal{C}$ such that $\mathsf{user}(\alpha) = \mathsf{false}$. Now consider the repair plan $(\mathcal{P}'_-, \mathcal{P}_+)$ where $\mathcal{P}'_-$ contains the following assertions

$$\mathcal{P}_- \cup \{\alpha \mid \exists \mathcal{C} \in \mathsf{causes}(q, \langle \mathcal{T}, \mathcal{A}\rangle), \alpha \in \mathcal{C}, \mathsf{user}(\alpha) = \mathsf{false}\}.$$

By construction, q is satisfied by the KB $\langle \mathcal{T}, (\mathcal{A} \setminus \mathcal{P}'_-) \cup \mathcal{P}_+\rangle$ induced by $(\mathcal{P}'_-, \mathcal{P}_+)$. Since $(\mathcal{P}_-, \mathcal{P}_+)$ is known to be validatable, and $\mathcal{P}'_- \setminus \mathcal{P}_- \subseteq False_{\mathsf{user}}$, it follows that $(\mathcal{P}'_-, \mathcal{P}_+)$ is also validatable. It follows from Lemma 5.2.5 that $\mathcal{S}_{\mathcal{W}}(\mathcal{P}_-, \mathcal{P}_+) \subseteq \mathcal{S}_{\mathcal{W}}(\mathcal{P}'_-, \mathcal{P}_+)$ and $\mathcal{S}_{\mathcal{U}}(\mathcal{P}_-, \mathcal{P}_+) \subseteq \mathcal{S}_{\mathcal{U}}(\mathcal{P}'_-, \mathcal{P}_+)$. We have thus found a validatable repair plan that extends $(\mathcal{P}_-, \mathcal{P}_+)$ and whose corresponding KB satisfies q and all answers that were already satisfied by $(\mathcal{P}_-, \mathcal{P}_+)$. We can therefore conclude that $q \in \mathcal{U}$ is satisfiable w.r.t. $(\mathcal{P}_-, \mathcal{P}_+)$. $\square$

Lemma 5.2.12. *Let $(\mathcal{P}_-, \mathcal{P}_+)$ be a validatable repair plan for the KB $\langle \mathcal{T}, \mathcal{A}\rangle$. Then an answer $q \in \mathcal{W}$ is satisfiable w.r.t. $(\mathcal{P}_-, \mathcal{P}_+)$ iff q is satisfiable for the KB $\langle \mathcal{T}, \mathcal{A}\rangle$ with a witness $\mathcal{C}_0$ such that every $q' \in \mathcal{S}_{\mathcal{U}}(\mathcal{P}_-, \mathcal{P}_+)$ is satisfiable for the KB $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+ \cup \mathcal{C}_0\rangle$.*

Proof. If $q \in \mathcal{W}$ is satisfiable w.r.t. $(\mathcal{P}_-, \mathcal{P}_+)$, then there exists a validatable repair plan $(\mathcal{P}'_-, \mathcal{P}'_+)$ such that $\mathcal{P}_- \subseteq \mathcal{P}'_-$ and $\mathcal{P}_+ \subseteq \mathcal{P}'_+$ which satisfies q and all answers in $\mathcal{S}(\mathcal{P}_-, \mathcal{P}_+)$. As q is satisfied by $(\mathcal{P}'_-, \mathcal{P}'_+)$, the ABox $(\mathcal{A} \setminus \mathcal{P}'_-) \cup \mathcal{P}'_+$ contains a cause $\mathcal{C}_0$ for q that has no conflict and that does not contain any false assertion. This means that q is satisfiable for $\langle \mathcal{T}, \mathcal{A}\rangle$. Now take some $q' \in \mathcal{S}_{\mathcal{U}}(\mathcal{P}_-, \mathcal{P}_+)$. Since $\mathcal{S}_{\mathcal{U}}(\mathcal{P}_-, \mathcal{P}_+) \subseteq \mathcal{S}_{\mathcal{U}}(\mathcal{P}'_-, \mathcal{P}'_+)$, we have $q' \in \mathcal{S}_{\mathcal{U}}(\mathcal{P}'_-, \mathcal{P}'_+)$, and so by Lemma 5.2.8, we have $\mathcal{P}'_- \cap \mathcal{C} \neq \emptyset$ for every $\mathcal{C} \in \mathsf{causes}(q', \langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}'_+\rangle)$. We observe that $\mathcal{P}'_- \subseteq False_{\mathsf{user}}$ and $\mathcal{A} \cup \mathcal{P}_+ \cup \mathcal{C}_0 \subseteq \mathcal{A} \cup \mathcal{P}'_+$. It follows that for every $\mathcal{C} \in \mathsf{causes}(q', \langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+ \cup \mathcal{C}_0\rangle)$, there exists $\alpha \in \mathcal{C}$ with $\mathsf{user}(\alpha) = \mathsf{false}$. By Lemma 5.2.9, we can conclude that q' is satisfiable for the KB $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+ \cup \mathcal{C}_0\rangle$.

In the other direction, suppose that $q \in \mathcal{W}$ is satisfiable for the KB $\langle \mathcal{T}, \mathcal{A}\rangle$ with a witness $\mathcal{C}_0$ such that every $q' \in \mathcal{S}_{\mathcal{U}}(\mathcal{P}_-, \mathcal{P}_+)$ is satisfiable for the KB $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+ \cup \mathcal{C}_0\rangle$. Consider the repair plan $(\mathcal{P}'_-, \mathcal{P}'_+)$ where

$$\mathcal{P}'_- = \mathcal{P}_- \cup \{\beta \in \mathcal{A} \mid \exists \alpha \in \mathcal{C}_0, \langle \mathcal{T}, \{\alpha, \beta\}\rangle \models \bot, \mathsf{user}(\beta) = \mathsf{false}\} \cup \{\alpha \mid \mathsf{user}(\alpha) = \mathsf{false}$$
$$\text{and there exists some } q' \in \mathcal{U} \text{ and } \mathcal{C} \in \mathsf{causes}(q', \langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+ \cup \mathcal{C}_0\rangle) \text{ such that } \alpha \in \mathcal{C}\}$$
$$\mathcal{P}'_+ = \mathcal{P}_+ \cup \{\alpha \in \mathcal{C}_0 \setminus \mathcal{A} \mid \mathsf{user}(\alpha) = \mathsf{true}\}$$

By construction, $(\mathcal{P}'_-, \mathcal{P}'_+)$ is validatable and satisfies q. We have $\mathcal{S}_{\mathcal{W}}(\mathcal{P}_-, \mathcal{P}_+) \subseteq \mathcal{S}_{\mathcal{W}}(\mathcal{P}'_-, \mathcal{P}'_+)$ by Lemma 5.2.5. To see why $\mathcal{S}_{\mathcal{U}}(\mathcal{P}_-, \mathcal{P}_+) \subseteq \mathcal{S}_{\mathcal{U}}(\mathcal{P}'_-, \mathcal{P}'_+)$, take some answer $q' \in \mathcal{S}_{\mathcal{U}}(\mathcal{P}_-, \mathcal{P}_+)$. By our earlier assumption, we know that q' is satisfiable for the KB $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+ \cup \mathcal{C}_0\rangle$, so by Lemma 5.2.9, every $\mathcal{C} \in \mathsf{causes}(q', \langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+ \cup \mathcal{C}_0\rangle)$ contains an assertion $\alpha \in \mathcal{C}$ such that $\mathsf{user}(\alpha) = \mathsf{false}$, which will thus be included in $\mathcal{P}'_-$. Since every cause for q' in $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+ \cup \mathcal{C}_0\rangle$ has a non-empty intersection with $\mathcal{P}'_-$, we can apply Lemma 5.2.8 to conclude that q' is satisfied by $(\mathcal{P}'_-, \mathcal{P}'_+)$. □

It follows from the preceding characterizations that deciding the satisfaction, satisfiability, or satisfiability w.r.t. a repair plan of an answer is tractable.

Proposition 5.2.13. *Deciding if an answer is satisfied, satisfiable, or satisfiable w.r.t. a repair plan is in* P.

Proof.
• Deciding if a wanted (resp. unwanted) answer is satisfied amounts to deciding if it is entailed under IAR semantics with a nonfalse explanation (resp. not entailed under brave semantics), so is in P w.r.t. $|\mathcal{A}|$.

• Since computing the causes of a query q is in P w.r.t. $|\mathcal{A}|$, and the number of causes is polynomial w.r.t. $|\mathcal{A}|$, the characterization of Lemma 5.2.9 shows that deciding if an unwanted answer is satisfiable is in P w.r.t. $|\mathcal{A}|$.

• Deciding if a wanted answer q is satisfiable using the characterization of Lemma 5.2.10 can be done by computing the causes of q and their conflicts in $\langle \mathcal{T}, \mathcal{A} \cup True^{rel}_{\mathsf{user}}\rangle$ in P w.r.t. $|\mathcal{A}|$ and $|True^{rel}_{\mathsf{user}}|$ and verifying in P that at least one of the causes fulfils the required conditions.

• By Lemma 5.2.11, checking whether $q \in \mathcal{U}$ is satisfiable w.r.t. $(\mathcal{P}_-, \mathcal{P}_+)$ reduces to checking whether $q \in \mathcal{U}$ is satisfiable for the KB $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+\rangle$. We know from earlier that the latter check can be done in P w.r.t. the size of the ABox. Since $\mathcal{P}_+ \subseteq True^{rel}_{\mathsf{user}}$, this condition can be verified in P w.r.t. $|\mathcal{A}|$ and $|True^{rel}_{\mathsf{user}}|$.

• To check whether an answer $q \in \mathcal{W}$ is satisfiable w.r.t. $(\mathcal{P}_-, \mathcal{P}_+)$, it suffices to check whether q satisfies the conditions of Lemma 5.2.12. These can be verified by: (i) computing the causes of q and their conflicts in $\langle \mathcal{T}, \mathcal{A} \cup True^{rel}_{\mathsf{user}}\rangle$, and (ii) for each candidate cause $\mathcal{C}_0$ that fulfils the conditions of Lemma 5.2.10, and every unwanted answer $q' \in U$, check that if q' is satisfied by $(\mathcal{P}_-, \mathcal{P}_+)$, then it is satisfiable for the KB $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+ \cup \mathcal{C}_0\rangle$. Everything can be done in P w.r.t. $|\mathcal{A}|$, $|True^{rel}_{\mathsf{user}}|$, and $|\mathcal{U}|$ with the same arguments as previous cases. □

We are now ready to establish the complexity of deciding the optimality of a repair plan. For the lower bounds, we will use the coNP-hard problems presented in the two following lemmas.

Lemma 5.2.14. NP*-hardness of SAT holds if we impose that at least one variable appears in positive and negative form in the formula.*

Proof. Reduction from SAT. Let $\{C_1,...,C_m\}$ be a set of clauses. $C_1 \wedge ... \wedge C_m$ is satisfiable iff $C_1 \wedge ... \wedge C_m \wedge (z \vee \neg z)$ is satisfiable, where z is a fresh variable. $\square$

Lemma 5.2.15. *The following problem is* NP*-hard: given a set $\{C_1,...,C_m,C_{m+1}\}$ of clauses such that $\{C_1,...,C_m\}$ is satisfiable and C_{m+1} is not a tautology: decide whether $\{C_1,...,C_m,C_{m+1}\}$ is satisfiable.*

Proof. Reduction from SAT. Let $\{C_1,...,C_m\}$ be a set of clauses. Then $C_1 \wedge ... \wedge C_m$ is satisfiable iff $(C_1 \vee \neg z) \wedge ... \wedge (C_m \vee \neg z) \wedge z$ is satisfiable, where z is a fresh variable. Clearly, $(C_1 \vee \neg z) \wedge ... \wedge (C_m \vee \neg z)$ is satisfiable and z is not a tautology. $\square$

Theorem 5.2.16. *Deciding if a repair plan is globally $\preceq$-optimal is co*NP*-complete for $\preceq \in \{\preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$, and in* P *for $\preceq \in \{\preceq_{\mathcal{W}}, \preceq_{\mathcal{U}}\}$. Deciding if a repair plan is locally $\preceq$-optimal is in* P *for $\preceq \in \{\preceq_{\mathcal{U}}, \preceq_{\mathcal{W}}, \preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$.*

Upper bounds. The tractability results follow from the characterizations of optimality given in Proposition 5.2.7 together with the complexity results of Proposition 5.2.13.

For the coNP upper bounds, we note that to show that $\mathcal{P}$ is *not* $\preceq$-optimal (for $\preceq \in \{\preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$), we can guess another repair plan $\mathcal{P}'$ and verify in P that both plans are validatable and that $\mathcal{P}'$ satisfies more answers than $\mathcal{P}$. $\square$

The lower bounds are as follows:

Lower bound for global $\preceq_{\mathcal{U},\mathcal{W}}$- and $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimality. Let Φ be a CNF formula of the form $\Phi = \bigwedge_{i=1}^{m+1} C_i$ over the variables $x_1,...,x_n$ such that $\bigwedge_{i=1}^{m} C_i$ is satisfiable and C_{m+1} is not a tautology (cf. Lemma 5.2.15). Consider the QRP defined as follows

$$\begin{aligned}
\mathcal{T} =& \{P \sqsubseteq S, N \sqsubseteq S\}\\
\mathcal{A} =& \{A(x_j), B(x_j) | 1 \leq j \leq n\} \cup \{P(b,x_j), N(b,x_j) \mid 1 \leq j \leq n\}\\
\mathcal{W} =& \{\exists x S(c_1,x),...,\exists x S(c_{m+1},x)\}\\
\mathcal{U} =& \{\exists xyz P(y,x) \wedge N(z,x) \wedge A(x) \wedge B(x)\}
\end{aligned}$$

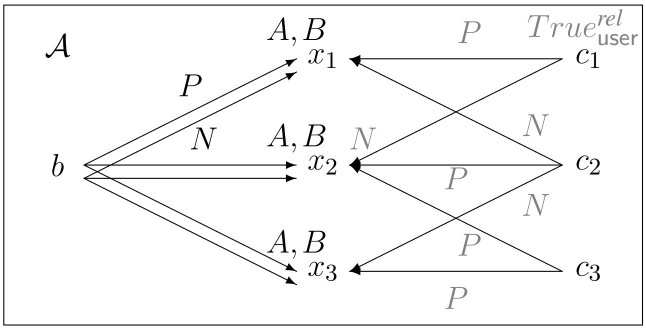
where

$$\begin{aligned}
True^{rel}_{\mathsf{user}} =& \{P(c_i,x_j) | x_j \in C_i\} \cup \{N(c_i,x_j) | \neg x_j \in C_i\}\\
False_{\mathsf{user}} =& \{P(b,x_j), N(b,x_j) \mid 1 \leq j \leq n\}\\
Unk_{\mathsf{user}} =& \mathcal{A} \setminus False_{\mathsf{user}}
\end{aligned}$$

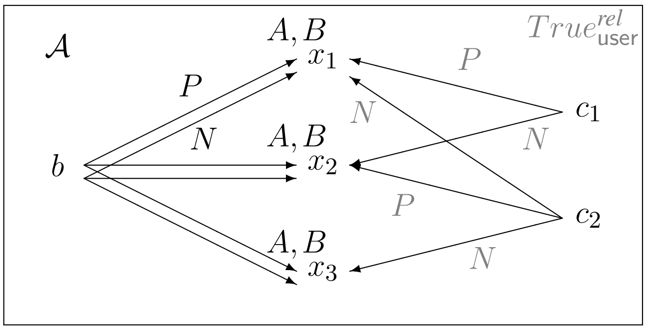
Let ν be a valuation of the x_j that satisfies $\bigwedge_{i=1}^{m} C_i$. We show that deciding if the repair $(\mathcal{P}_-, \mathcal{P}_+)$ with

$$\begin{aligned}
\mathcal{P}_- =& False_{\mathsf{user}}\\
\mathcal{P}_+ =& \{P(c_i,x_j) \mid x_j \in C_i, \nu(x_j) = \mathsf{true}, 1 \leq i \leq m\} \cup\\
& \{N(c_i,x_j) \mid \neg x_j \in C_i, \nu(x_j) = \mathsf{false}, 1 \leq i \leq m\}
\end{aligned}$$

Fig. 5.1 Reduction for coNP-hardness of recognition of globally $\preceq_{\mathcal{U},\mathcal{W}}$ or $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal optimal repair plans. Graphical representation of $\mathcal{A} \cup True^{rel}_{\mathsf{user}}$ constructed from an example set of clauses $\Phi = \{C_1 = x_1 \vee \neg x_2, C_2 = \neg x_1 \vee x_2 \vee \neg x_3\} \cup \{C_3 = x_2 \vee x_3\}$.



Reduction for coNP-hardness of recognition of globally $\preceq_{\mathcal{W},\mathcal{U}}$-optimal optimal repair plans. Graphical representation of $\mathcal{A} \cup True^{rel}_{\mathsf{user}}$ constructed from an example set of clauses $\Phi = \{C_1 = x_1 \vee \neg x_2, C_2 = \neg x_1 \vee x_2 \vee \neg x_3\}$.



is not globally $\preceq_{\mathcal{U},\mathcal{W}}$-optimal iff Φ is satisfiable.

First observe that $(\mathcal{P}_-,\mathcal{P}_+)$ is validatable and satisfies the single unwanted answer. Moreover, as ν satisfies the clauses $c_1,\ldots,c_m$, all of the wanted answers concerning the individuals $c_1,...,c_m$ are satisfied by $(\mathcal{P}_-,\mathcal{P}_+)$.

If Φ is satisfiable, let ν' be a valuation of the x_j that satisfies Φ. It is readily verified that the repair plan $(\mathcal{P}'_-,\mathcal{P}'_+)$ with

$$\begin{aligned}\mathcal{P}'_- =& False_{\mathsf{user}}\\ \mathcal{P}'_+ =& \{P(c_i,x_j) \mid x_j \in C_i, \nu'(x_j) = \mathsf{true}, 1 \leq i \leq m+1\} \cup\\ & \{N(c_i,x_j) \mid \neg x_j \in C_i, \nu'(x_j) = \mathsf{false}, 1 \leq i \leq m+1\}\end{aligned}$$

is validatable and satisfies all unwanted and wanted answers, so $(\mathcal{P}_-,\mathcal{P}_+)$ is not $\preceq_{\mathcal{U},\mathcal{W}}$-globally optimal.

In the other direction, if $(\mathcal{P}_-,\mathcal{P}_+)$ is not globally $\preceq_{\mathcal{U},\mathcal{W}}$-optimal, then there must exist a repair plan $(\mathcal{P}'_-,\mathcal{P}'_+)$ that is validatable and satisfies all of the answers in $\mathcal{U}\cup\mathcal{W}$. Then it can be straightforwardly verified that Φ is satisfied by the valuation ν' of the x_j defined by $\nu'(x_j) = \mathsf{true}$ iff there exists c_i such that $P(c_i,x_j) \in \mathcal{P}_+$. Indeed, every c_i has an outgoing edge in $(\mathcal{A}\backslash\mathcal{P}'_-)\cup\mathcal{P}'_+$, and no x_j has both P- and N- incoming edges, since otherwise the unwanted answer would not be satisfied. $\square$

Lower bound for global $\preceq_{\mathcal{W},\mathcal{U}}$-optimality. The proof is by reduction from SAT when at least one variable appears both in positive and negative form in the formula. Take some CNF formula $\Phi = \bigwedge_{i=1}^m C_i$ over the variables $x_1,...,x_n$ that satisfies this requirement, and consider the QRP defined as follows:

$$\begin{aligned}\mathcal{T} =& \{P \sqsubseteq S, N \sqsubseteq S\}\\ \mathcal{A} =& \{A(x_j), B(x_j) | 1 \leq j \leq n\} \cup \{P(b,x_j), N(b,x_j) \mid 1 \leq j \leq n\}\\ \mathcal{W} =& \{\exists x S(c_1,x),...,\exists x S(c_m,x)\}\\ \mathcal{U} =& \{\exists xyz P(y,x) \wedge N(z,x) \wedge A(x) \wedge B(x)\}\end{aligned}$$

where:

$$\begin{aligned}True^{\mathit{rel}}_{\mathsf{user}} =& \{P(c_i,x_j) | x_j \in C_i\} \cup \{N(c_i,x_j) | \neg x_j \in C_i\}\\ False_{\mathsf{user}} =& \{P(b,x_j), N(b,x_j) \mid 1 \leq j \leq n\}\\ Unk_{\mathsf{user}} =& \mathcal{A}\backslash False_{\mathsf{user}}\end{aligned}$$

It is easy to see that the repair plan $(\mathcal{P}_-,\mathcal{P}_+) = (False_{\mathsf{user}}, True^{\mathit{rel}}_{\mathsf{user}})$ is validatable and satisfies all wanted answers but does not satisfy the unwanted answer because at least one x_j has both incoming N- and P-edges. In fact, we can show that $(\mathcal{P}_-,\mathcal{P}_+)$ is not globally $\preceq_{\mathcal{W},\mathcal{U}}$-optimal (i.e., there is some validatable repair plan that satisfies all of $\mathcal{U}\cup\mathcal{W}$) iff Φ is satisfiable. Indeed, every validatable repair plan that satisfies all unwanted and wanted answers gives rise to a satisfying valuation for Φ, and conversely, any such valuation induces such a repair plan (add either $N(c_i, x_j)$ or $P(c_i, x_j)$ for each x_j in such a way that every c_i has an outgoing edge). $\square$

Remark 5.2.17. If $\mathcal{U}$ contains only instance queries, deciding if a repair plan is globally $\preceq$-optimal for $\preceq \in \{\preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$ is in P. Indeed, because of the truthfulness condition, every assertion that entails some $q \in \mathcal{U}$ is false, so cannot be used to satisfy a wanted answer. There is therefore no need to make the compromise between the satisfaction of wanted and unwanted answers. A repair plan $(\mathcal{P}_-, \mathcal{P}_+)$ is globally $\preceq$-optimal iff it is globally $\preceq_{\mathcal{U}}$-optimal (which is in this case equivalent to $\mathcal{P}_-$ contains all causes of unwanted answers) and globally $\preceq_{\mathcal{W}}$-optimal.

5.2.2 Generic algorithms

Our complexity analysis reveals that the notions of global optimality based upon the preference relations $\preceq_{\{\mathcal{U},\mathcal{W}\}}$, $\preceq_{\mathcal{U},\mathcal{W}}$, and $\preceq_{\mathcal{W},\mathcal{U}}$ have undesirable computational properties: even when provided with all relevant user knowledge, it is intractable to decide whether a given plan is optimal. Moreover, while plans globally $\preceq_{\mathcal{U}}$- (resp. $\preceq_{\mathcal{W}}$-) optimal can be interactively constructed in a monotonic fashion by removing further false assertions (resp. and adding further true assertions), building a globally optimal plan for a preference relation that involves both $\mathcal{U}$ and $\mathcal{W}$ may require backtracking over answers already satisfied (cf. the situation in Example 5.2.4). We therefore target validatable repair plans that are both *globally optimal for $\preceq_{\mathcal{U}}$ or $\preceq_{\mathcal{W}}$* (depending which is preferred) and *locally optimal for $\preceq_{\{\mathcal{U},\mathcal{W}\}}$*.

Remark 5.2.18. The repair plans we target actually cover some other cases. Indeed, it follows from Proposition 5.2.7 that the following notions of optimality are equivalent:

global $\preceq_{\mathcal{U}}$- and local $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimality $\iff$ global $\preceq_{\mathcal{U}}$- and local $\preceq_{\mathcal{U},\mathcal{W}}$-optimality
global $\preceq_{\mathcal{W}}$- and local $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimality $\iff$ global $\preceq_{\mathcal{W}}$- and local $\preceq_{\mathcal{W},\mathcal{U}}$-optimality
$\iff$ global $\preceq_{\mathcal{W}}$- and local $\preceq_{\mathcal{U}}$-optimality
$\iff$ local $\preceq_{\mathcal{W},\mathcal{U}}$-optimality

It is also notable that for our target repair plans, it is equivalent to replace set-inclusion by cardinality for the comparison between sets of satisfied answers in the definition of the preorders. In the following we use $\preceq^{\leq}$ and $\preceq^{\subseteq}$ to denote these two families of preorders. We show that being globally (resp. locally) optimal for $\preceq^{\leq}$ and $\preceq^{\subseteq}$ is equivalent for $\preceq \in \{\preceq_{\mathcal{U}}, \preceq_{\mathcal{W}}\}$ (resp. $\preceq \in \{\preceq_{\mathcal{U}}, \preceq_{\mathcal{W}}, \preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$).

Proposition 5.2.19. *For $\preceq \in \{\preceq_{\mathcal{U}}, \preceq_{\mathcal{W}}\}$, a repair plan is globally $\preceq^{\leq}$-optimal iff it is globally $\preceq^{\subseteq}$-optimal.*

Proof. A repair plan is globally $\preceq^{\subseteq}_{\mathcal{U}}$-optimal (resp. $\preceq^{\subseteq}_{\mathcal{W}}$-optimal) iff it satisfies all satisfiable unwanted (resp. wanted) answers, so iff it is globally $\preceq^{\leq}_{\mathcal{U}}$-optimal (resp. $\preceq^{\leq}_{\mathcal{W}}$-optimal). $\square$

Remark 5.2.20. Even for the other preorders, the complexity of recognizing a globally optimal repair plan is the same when preorders are defined using cardinality: to show that a repair plan is not optimal, it is still possible to guess a better one and check in P that it is actually better.

Proposition 5.2.21. *For $\preceq \in \{\preceq_{\mathcal{U}}, \preceq_{\mathcal{W}}, \preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$, a repair plan is locally $\preceq^{\leq}$-optimal iff it is locally $\preceq^{\subseteq}$-optimal.*

Proof. Let $\mathcal{P} = (\mathcal{P}_-, \mathcal{P}_+)$ be a locally $\preceq^{\subseteq}$-optimal repair plan and suppose for a contradiction that there exists $\mathcal{P}' = (\mathcal{P}'_-, \mathcal{P}'_+)$ such that $\mathcal{P}_- \subseteq \mathcal{P}'_-$, $\mathcal{P}_+ \subseteq \mathcal{P}'_+$ and $\mathcal{P} \prec^{\leq} \mathcal{P}'$.

Since $\mathcal{P}_- \subseteq \mathcal{P}'_-$ and $\mathcal{P}_+ \subseteq \mathcal{P}'_+$, then by Lemma 5.2.5 $\mathcal{S}_{\mathcal{W}}(\mathcal{P}) \subseteq \mathcal{S}_{\mathcal{W}}(\mathcal{P}')$, so if $|\mathcal{S}_{\mathcal{W}}(\mathcal{P})| < |\mathcal{S}_{\mathcal{W}}(\mathcal{P}')|$, then $\mathcal{S}_{\mathcal{W}}(\mathcal{P}) \subset \mathcal{S}_{\mathcal{W}}(\mathcal{P}')$.

It follows that if $\preceq = \preceq_{\mathcal{W}}$, $\mathcal{P} \prec^{\leq} \mathcal{P}'$ implies $\mathcal{P} \prec^{\subseteq} \mathcal{P}'$, so yields a contradiction.

Moreover, if $\preceq \neq \preceq_{\mathcal{W}}$, $\mathcal{P}$ satisfies all unwanted answers that are satisfiable w.r.t. $\mathcal{P}$, and it is not possible that $\mathcal{P}'$ satisfies some unwanted answers that are not satisfiable w.r.t. $\mathcal{P}$, because those answers have some causes in $\mathcal{A} \cup \mathcal{P}_+$, so also in $\mathcal{A} \cup \mathcal{P}'_+$, which do not contain any false assertions, so $\mathcal{S}_{\mathcal{U}}(\mathcal{P}') \subseteq \mathcal{S}_{\mathcal{U}}(\mathcal{P})$.

It follows that if $\preceq \in \{\preceq_{\mathcal{U}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\{\mathcal{U},\mathcal{W}\}}\}$, then $|\mathcal{S}_{\mathcal{U}}(\mathcal{P})| \leq |\mathcal{S}_{\mathcal{U}}(\mathcal{P}')|$, so $\mathcal{S}_{\mathcal{U}}(\mathcal{P}') = \mathcal{S}_{\mathcal{U}}(\mathcal{P})$. Then, since $\mathcal{P} \prec^{\leq} \mathcal{P}'$, $|\mathcal{S}_{\mathcal{W}}(\mathcal{P})| < |\mathcal{S}_{\mathcal{W}}(\mathcal{P}')|$, so $|\mathcal{S}_{\mathcal{W}}(\mathcal{P})| \subset |\mathcal{S}_{\mathcal{W}}(\mathcal{P}')|$, and $\mathcal{P} \prec^{\subseteq} \mathcal{P}'$, which contradicts the local $\preceq^{\subseteq}$-optimality of $\mathcal{P}$.

In the case $\preceq = \preceq_{\mathcal{W},\mathcal{U}}$, either $|\mathcal{S}_{\mathcal{W}}(\mathcal{P})| < |\mathcal{S}_{\mathcal{W}}(\mathcal{P}')|$ and $\mathcal{S}_{\mathcal{W}}(\mathcal{P}) \subset \mathcal{S}_{\mathcal{W}}(\mathcal{P}')$, which contradicts the local $\preceq^{\subseteq}$-optimality of $\mathcal{P}$, or $\mathcal{S}_{\mathcal{W}}(\mathcal{P}) = \mathcal{S}_{\mathcal{W}}(\mathcal{P}')$ and $|\mathcal{S}_{\mathcal{U}}(\mathcal{P})| < |\mathcal{S}_{\mathcal{U}}(\mathcal{P}')|$, which is impossible because $\mathcal{S}_{\mathcal{U}}(\mathcal{P}') \subseteq \mathcal{S}_{\mathcal{U}}(\mathcal{P})$. □

In Algorithm 5.6, we give an interactive algorithm $\mathsf{OptimalRepairPlan}_{\mathcal{U}}$ for building a repair plan that is both globally optimal for $\preceq_{\mathcal{U}}$ and locally optimal for $\preceq_{\{\mathcal{U},\mathcal{W}\}}$ to address the case where $\mathcal{U}$ is preferred; if $\mathcal{W}$ is preferred, we use the algorithm $\mathsf{OptimalRepairPlan}_{\mathcal{W}}$ obtained by removing Step 14 from $\mathsf{OptimalRepairPlan}_{\mathcal{U}}$ to construct a repair plan that is globally optimal for $\preceq_{\mathcal{W}}$ and locally optimal for $\preceq_{\{\mathcal{U},\mathcal{W}\}}$. The idea of $\mathsf{OptimalRepairPlan}_{\mathcal{U}}$ is to remove every false assertion involved in the QRP in order to satisfy the unwanted answers and wanted answers that are satisfiable by removing assertions, then to try to satisfy the wanted answers that are not satisfied by adding assertions, while preserving the satisfied unwanted answers. The algorithms terminate provided the user knows only a finite number of assertions that may be inserted. In this case, the algorithms output optimal repair plans:

Theorem 5.2.22. *The output of* $\mathsf{OptimalRepairPlan}_{\mathcal{U}}$ *(resp.* $\mathsf{OptimalRepairPlan}_{\mathcal{W}}$*) is globally $\preceq_{\mathcal{U}}$ (resp. $\preceq_{\mathcal{W}}$) and locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal.*

Proof. We give first the proof for $\mathsf{OptimalRepairPlan}_{\mathcal{U}}$.

First observe that at every point during the execution of the algorithm, the current repair plan is validatable, since only true assertions are added to $\mathcal{P}_+$ and false assertions are added to $\mathcal{P}_-$ (they are either marked as false by the user, or conflict with assertions that have been marked as true).

Step 2 adds to $\mathcal{P}_-$ all assertions known to be false that belong to a cause of some $q \in \mathcal{U} \cup \mathcal{W}$ or a conflict of some cause of $q \in \mathcal{W}$. Thus, at the end of this step, $\mathcal{P}_-$ satisfies every satisfiable answer in $\mathcal{U}$, that is, every answer in $\mathcal{U}$ every cause of which contains at least one false assertion (cf. proof of Proposition 5.2.13). Hence $(\mathcal{P}_-, \mathcal{P}_+)$ is globally $\preceq_{\mathcal{U}}$-optimal at the end of Step 2. Moreover, every false assertion that occurs in a cause or conflict of a cause of a wanted answer has been removed, so if $q \in \mathcal{W}$ is not satisfied at this point, then it has no cause without any conflict in $\mathcal{A} \backslash \{\alpha \mid \mathsf{user}(\alpha) = \mathsf{false}\}$.

Algorithm 5.6 $\mathsf{OptimalRepairPlan}_{\mathcal{U}}$

Input: QRP $(\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle, \mathcal{U}, \mathcal{W})$

Output: repair plan

1: $\mathcal{P}_- \leftarrow \emptyset, \mathcal{P}_+ \leftarrow \emptyset$

2: Display the assertions of $\bigcup_{q \in \mathcal{U} \cup \mathcal{W}} \mathsf{causes}(q, \mathcal{K})$ and $\bigcup_{q \in \mathcal{W}, \mathcal{C} \in \mathsf{causes}(q,\mathcal{K})} \mathsf{confl}(\mathcal{C}, \mathcal{K})$

3: Ask user to mark *all* false (F) and true (T) assertions

4: $\mathcal{P}_- \leftarrow \mathcal{P}_- \cup F \cup \mathsf{confl}(T, \mathcal{K})$

5: **while** $\mathcal{W}' = \mathcal{W} \setminus \mathcal{S}_{\mathcal{W}}(\mathcal{P}_-, \mathcal{P}_+) \neq \emptyset$ **do**

6:     $q \leftarrow$ first$(\mathcal{W}')$

7:     Ask the user for true assertions T_q (not already provided) to complete (or create) a cause for q

8:     **if** $T_q = \emptyset$ **then** // nothing to add, q cannot be satisfied

9:         $\mathcal{W} \leftarrow \mathcal{W} \setminus \{q\}$, go to 5 // try to satisfy next unsatisfied wanted answer

10:     **end if**

11:     $\mathcal{P}_+ \leftarrow \mathcal{P}_+ \cup T_q$, $\mathcal{P}_- \leftarrow \mathcal{P}_- \cup \mathsf{confl}(T_q, \langle \mathcal{T}, \mathcal{A} \cup T_q \rangle)$

12:     Show assertions of every cause $\mathcal{C}$ of q such that $T_q \cap \mathcal{C} \neq \emptyset$ and its conflicts: user indicates *all* false, true assertions F', T': $\mathcal{P}_- \leftarrow \mathcal{P}_- \cup F' \cup \mathsf{confl}(T', \mathcal{K})$

13:     Show assertions of causes of every $q' \in \mathcal{U}$ in $\mathcal{A} \setminus \mathcal{P}_- \cup \mathcal{P}_+$: user indicates *all* false assertions F'': $\mathcal{P}_- \leftarrow \mathcal{P}_- \cup F''$

14:     **if** there is $q'' \in \mathcal{U}$ such that $\langle \mathcal{T}, \mathcal{A} \setminus \mathcal{P}_- \cup \mathcal{P}_+ \rangle \models_{\text{brave}} q''$ and $(\mathcal{T}, \mathcal{A} \setminus \mathcal{P}_-) \not\models_{\text{brave}} q''$ **then**

15:         $\mathcal{P}_+ \leftarrow \mathcal{P}_+ \setminus T_q$ // revert $\mathcal{P}_+$

16:     **end if**

17: **end while**

18: Return $(\mathcal{P}_-, \mathcal{P}_+)$

The purpose of Step 5 is to add new true assertions to create causes for the wanted answers not satisfied after Step 2, while preserving $\mathcal{S}_{\mathcal{U}}(\mathcal{P}_-,\mathcal{P}_+)$. For every $q \in \mathcal{W}$, while q is not satisfied, the user is asked to input true assertions to complete a cause for q in Step 7. If he is unable to do so, at Step 8, we remove q from $\mathcal{W}$ (since it cannot be satisfied w.r.t. user); otherwise, we update $\mathcal{P}_-$ and $\mathcal{P}_+$ using T_q (Step 11). Note that since T_q contains only true assertions, we can remove its conflicts without affecting already satisfied wanted answers; this step is necessary because T_q may conflict with assertions of $\mathcal{A}$ that are not involved in the causes and conflicts presented at Step 2. In Step 12, we remove false assertions appearing in a new cause for q or its conflicts (such assertions may not have been examined in Step 2). Step 13 removes false assertions of new causes of unwanted answers, and Step 14 undoes the addition of T_q if it affects $\mathcal{S}_{\mathcal{U}}(\mathcal{P}_-,\mathcal{P}_+)$. Thus, at the end of Step 5, for every wanted answer, either it is satisfied, or the user is unable to supply a cause that does not deteriorate $\mathcal{S}_{\mathcal{U}}(\mathcal{P}_-,\mathcal{P}_+)$. It follows that $(\mathcal{P}_-,\mathcal{P}_+)$ is locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal.

For $\mathsf{OptimalRepairPlan}_{\mathcal{W}}$, Step 14 is removed, so every satisfiable answer in $\mathcal{W}$ is satisfied at the end of Step 5, and $(\mathcal{P}_-,\mathcal{P}_+)$ is globally $\preceq_{\mathcal{W}}$-optimal. To see why $(\mathcal{P}_-,\mathcal{P}_+)$ is locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal, observe that $(\mathcal{P}_-,\mathcal{P}_+)$ satisfies every $q \in \mathcal{U}$ that is satisfiable w.r.t. $(\mathcal{P}_-,\mathcal{P}_+)$, i.e. is such that every cause for q in $\mathcal{A} \cup \mathcal{P}_+$ contains some false assertion. Indeed, the assertions of every such cause have been presented to the user either at Step 2 or at Step 13. □

5.3 Optimal deletion-only repair plans

In this section, we restrict our attention to constructing optimal deletion-only repair plans. In this simpler setting, all of the previously introduced notions of optimality collapse into the one characterized in the following proposition.

Proposition 5.3.1. *A validatable deletion-only plan is* optimal *iff it satisfies every $q \in \mathcal{U}$ such that every $\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})$ has $\alpha \in \mathcal{C}$ with $\mathsf{user}(\alpha) = \mathsf{false}$, and every $q \in \mathcal{W}$ for which there exists $\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})$ such that $\mathsf{user}(\alpha) \neq \mathsf{false}$ for every $\alpha \in \mathcal{C}$ and $\mathsf{user}(\beta) = \mathsf{false}$ for every $\beta \in \mathsf{confl}(\mathcal{C},\mathcal{K})$.*

Proof. In the case of deletion-only repair plans, being satisfiable or satisfiable w.r.t. a given repair plan is equivalent since removing more false assertions can only improve the satisfied answers (see Lemma 5.2.5). Hence, a validatable deletion-only plan is optimal iff it satisfies every answer satisfied by the "maximal" deletion-only plan $\{\alpha \mid \mathsf{user}(\alpha) = \mathsf{false}\}$, or more precisely: every $q \in \mathcal{U}$ such that every $\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})$ has $\alpha \in \mathcal{C}$ with $\mathsf{user}(\alpha) = \mathsf{false}$, and every $q \in \mathcal{W}$ for which there exists $\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})$ such that $\mathsf{user}(\alpha) \neq \mathsf{false}$ for every $\alpha \in \mathcal{C}$ and $\mathsf{user}(\beta) = \mathsf{false}$ for every $\beta \in \mathsf{confl}(\mathcal{C},\mathcal{K})$. □

Constructing such repair plans can be done with one of the preceding algorithms, omitting Step 5 that adds facts. However, it is possible to further assist the user by taking advantage of the fact that subsets of the ABox whose removal addresses all defects of the QRP can be automatically identified, and then interactively transformed into optimal repair plans. We call such subsets *potential solutions*.

Fig. 5.2 SAT encoding for potential solutions.

$$\varphi_{\mathcal{U}} = \bigwedge_{q\in\mathcal{U}} \bigwedge_{\mathcal{C}\in\mathsf{causes}(q,\mathcal{K})} \bigvee_{\alpha\in\mathcal{C}} x_\alpha$$

$$\begin{aligned}\varphi_{\mathcal{W}} = & \bigwedge_{q\in\mathcal{W}} \bigvee_{\mathcal{C}\in\mathsf{causes}(q,\mathcal{K})} w_{\mathcal{C}} \\ & \wedge \bigwedge_{q\in\mathcal{W}} \bigwedge_{\mathcal{C}\in\mathsf{causes}(q,\mathcal{K})} \bigwedge_{\alpha\in\mathcal{C}} \neg w_{\mathcal{C}} \vee \neg x_\alpha \\ & \wedge \bigwedge_{q\in\mathcal{W}} \bigwedge_{\mathcal{C}\in\mathsf{causes}(q,\mathcal{K})} \bigwedge_{\beta\in\mathsf{confl}(\mathcal{C},\mathcal{K})} \neg w_{\mathcal{C}} \vee x_\beta\end{aligned}$$

An assertion is said to be *relevant* if it appears in a cause of some $q \in \mathcal{U} \cup \mathcal{W}$ or in the conflicts of a cause of some $q \in \mathcal{W}$. If an assertion α appears in every potential solution, either $\mathsf{user}(\alpha) = \mathsf{false}$, or there is no validatable potential solution. We call such assertions *necessarily false*. If α appears in no potential solution, it is necessary to keep it in $\mathcal{A}$ to retrieve some wanted answers under IAR semantics, so either $\mathsf{user}(\alpha) \neq \mathsf{false}$, or it is not possible to satisfy all wanted answers. We call such assertions *necessarily nonfalse*.

When a potential solution does not exist, a *minimal correction subset of wanted answers* (MCSW) is an inclusion-minimal subset $\mathcal{W}' \subseteq \mathcal{W}$ such that removing $\mathcal{W}'$ from $\mathcal{W}$ yields a QRP with a potential solution. Because of the truthfulness condition, we know that the absence of a potential solution means that some wanted answers are supported only by causes containing erroneous assertions (otherwise the wanted and unwanted answers would be contradictory, which would violate the truthfulness condition). Moreover, since removing all such answers from $\mathcal{W}$ yields the existence of a potential solution, there exists a MCSW which contains only such answers, which we call an *erroneous MCSW*. This is why MCSWs can help identify the wanted answers that cannot be satisfied by a deletion-only repair plan. We will present an algorithm that exploits necessarily (non)false assertions and MCSWs to help the user in constructing an optimal deletion-only repair plan.

5.3.1 SAT encoding and complexity results

We first give a propositional encoding that can be used to compute the necessarily (non)false assertions or the MCSWs. We will use the *minimal correction subsets* of a formula.

Definition 5.3.2 (Minimal Correction Subset)**.** Given sets F and H of soft and hard clauses respectively, a subset $M \subseteq S$ is a *minimal correction subset* (MCS) of S w.r.t. H if (i) $(S\backslash M) \cup H$ is satisfiable, and (ii) $(S\backslash M') \cup H$ is unsatisfiable for every $M' \subsetneq M$

We construct in polynomial time the propositional CNF $\varphi = \varphi_{\mathcal{U}} \wedge \varphi_{\mathcal{W}}$, where $\varphi_{\mathcal{U}}$ and $\varphi_{\mathcal{W}}$ are defined in Figure 5.2, and rely on the results of the following lemma:

Lemma 5.3.3. *The CNF formula φ has the following properties:*

- *there exists a potential solution iff φ is satisfiable (every satisfying assignment corresponds to a potential solution);*
- *α is necessarily false iff $\varphi \wedge \neg x_\alpha$ is unsatisfiable;*
- *α is necessarily nonfalse iff $\varphi \wedge x_\alpha$ is unsatisfiable;*
- *let $H = \{\bigvee_{\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})} w_\mathcal{C} \mid q \in \mathcal{W}\}$, and $S = \varphi \backslash H$, the MCSWs correspond to the MCSs of S w.r.t. H.*

Proof. First suppose that there exists a potential solution $\mathcal{P}$, and let ν be a valuation of the variables of φ defined as follows: $\nu(x_\alpha) = \mathsf{true}$ iff $\alpha \in \mathcal{P}$, and $\nu(w_\mathcal{C}) = \mathsf{true}$ iff $\mathcal{C} \subseteq \mathcal{A} \backslash \mathcal{P}$ and $\mathsf{confl}(\mathcal{C},\mathcal{K}) \subseteq \mathcal{P}$ for every $\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})$ with $q \in \mathcal{W}$.

Since $\mathcal{P}$ is a potential solution, it contains at least one assertion of each cause of every unwanted answer, otherwise this answer would still be entailed under brave semantics in $\mathcal{A} \backslash \mathcal{P}$. It follows that $\varphi_\mathcal{U}$ is satisfied by ν. Moreover, every $q \in \mathcal{W}$ has at least one cause $\mathcal{C}$ without any conflict in $\mathcal{A} \backslash \mathcal{P}$, so $\mathcal{C} \cap \mathcal{P} = \emptyset$ and $\mathsf{confl}(\mathcal{C},\mathcal{K}) \subseteq \mathcal{P}$. By the way we defined ν, it satisfies $\varphi_\mathcal{W}$, and hence the full formula φ.

In the other direction, suppose that the formula φ is satisfiable, with satisfying valuation ν. Let $\mathcal{P} = \{\alpha \mid \nu(x_\alpha) = \mathsf{true}\}$. For every $q \in \mathcal{U}$ and $\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})$, $\mathcal{P}$ contains an assertion $\alpha \in \mathcal{C}$, so there is no cause for q in $\mathcal{A} \backslash \mathcal{P}$, so every $q \in \mathcal{U}$ is satisfied by $\mathcal{P}$. For every $q \in \mathcal{W}$, there is a cause $\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})$ such that $\nu(w_\mathcal{C}) = \mathsf{true}$. By the way we defined φ, this means that for every $\alpha \in \mathcal{C}$, $\nu(\alpha) = \mathsf{false}$, so $\mathcal{C} \cap \mathcal{P} = \emptyset$, and for every $\beta \in \mathsf{confl}(\mathcal{C},\mathcal{K})$, $\nu(\beta) = \mathsf{true}$, so $\mathsf{confl}(\mathcal{C},\mathcal{K}) \subseteq \mathcal{P}$. It follows that all $q \in \mathcal{W}$ are satisfied by $\mathcal{P}$.

Since the assertions assigned to true in a satisfying assignment correspond to a potential solution, α is necessarily false (resp. necessarily nonfalse) iff $\varphi \wedge \neg x_\alpha$ (resp. $\varphi \wedge x_\alpha$) is unsatisfiable: α belongs to every potential solution (resp. no potential solution) iff there is no satisfying valuation with α assigns to false (resp. to true).

For the final point, we will show that $\mathcal{M} \subseteq \mathcal{W}$ is a MCSW iff $M = \{\bigvee_{\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})} w_\mathcal{C} \mid q \in \mathcal{M}\}$ is a MCS of S w.r.t. H. First suppose that $\mathcal{M}$ is a MCSW. Since removing $\mathcal{M}$ from $\mathcal{W}$ yields a QRP that has a potential solution $\mathcal{P}$, the valuation ν such that $\nu(w_\mathcal{C}) = \mathsf{false}$ for every $\mathcal{C} \in \mathsf{causes}(q)$ with $q \in \mathcal{M}$, and $\nu(x_\alpha) = \mathsf{true}$ iff $\alpha \in \mathcal{P}$, and $\nu(w_\mathcal{C}) = \mathsf{true}$ iff $\mathcal{C} \subseteq \mathcal{A} \backslash \mathcal{P}$ and $\mathsf{confl}(\mathcal{C},\mathcal{K}) \subseteq \mathcal{P}$ for $\mathcal{C} \in \mathsf{causes}(q)$ with $q \in \mathcal{W} \backslash \mathcal{M}$ satisfies $\varphi \backslash \mathcal{M}$. Moreover, since removing $\mathcal{M}' \subsetneq \mathcal{M}$ from $\mathcal{W}$ does not yield a QRP that has a potential solution, M is a MCS. The other direction is similar. $\square$

To show that this approach is optimal from the complexity point of view, we establish the complexity of deciding if a potential solution exists, if an assertion is necessarily (non)false, and if $\mathcal{W}' \subseteq \mathcal{W}$ is a MCSW. For the lower bounds, we need the following lemma.

Lemma 5.3.4. *Given two sets of soft and hard clauses S, H, deciding if $M \subseteq S$ is a MCS of S w.r.t. H is BH_2-complete.*

Proof. To show that M is a MCS of S: show in NP that $(S \backslash M) \cup H$ is satisfiable and in coNP that M is minimal (to show in NP that M is not minimal, guess $M' \subseteq M$ and a valuation that satisfies $(S \backslash M') \cup H$).

Hardness is shown by reduction from SAT-UNSAT: let φ_S, φ_U be two CNF formulas that do not share variables. Then $\neg x$ is a MCS of $\varphi = \varphi_S \wedge (\varphi_U \vee x) \wedge \neg x$ iff φ_S is satisfiable and φ_U is unsatisfiable. $\square$

Theorem 5.3.5. *For complexity w.r.t. $|\mathcal{A}|$, $|\mathcal{U}|$ and $|\mathcal{W}|$, deciding if a set of assertions of $\mathcal{A}$ is a potential solution is in* P*, deciding if a potential solution exists is* NP*-complete, deciding if an assertion is necessarily (non)false is co*NP*-complete, and deciding if $\mathcal{W}' \subseteq \mathcal{W}$ is a MCSW is* BH_2*-complete.*

Proof. Deciding if $\mathcal{P} \subseteq \mathcal{A}$ is a potential solution amounts to verifying that $\langle \mathcal{T}, \mathcal{A} \backslash \mathcal{P} \rangle \models_{\mathrm{IAR}} q$ for every $q \in \mathcal{W}$, and $\langle \mathcal{T}, \mathcal{A} \backslash \mathcal{P} \rangle \not\models_{\mathsf{brave}} q$ for every $q \in \mathcal{U}$.

The other upper bounds follow from Lemma 5.3.3 and the fact that the formula φ can be constructed in polynomial time in $|\mathcal{A}|$, $|\mathcal{U}|$ and $|\mathcal{W}|$. Indeed, the construction relies upon computing the causes and conflicts of (un)wanted answers, which is known to be computable in P in $|\mathcal{A}|$.

The lower bounds can be shown by reduction from propositional satisfiability related problems. Figure 5.3 illustrates the different reductions.

Existence: The proof is by reduction from satisfiability of a CNF $C_1 \wedge ... \wedge C_m$ over $x_1, ...x_n$. Consider the following QRP setting:

$$\begin{aligned}
\mathcal{T}_0 =& \{\exists P \sqsubseteq S, \exists N \sqsubseteq S\} \\
\mathcal{A}_0 =& \{P(c_i, x_j) | x_j \in C_i\} \cup \{N(c_i, x_j) | \neg x_j \in C_i\} \\
\mathcal{W}_0 =& \{S(c_1), ..., S(c_m)\} \\
\mathcal{U} =& \{\exists x, y, z P(x, y) \wedge N(z, y)\}
\end{aligned}$$

We show that there exists a potential solution iff $C_1 \wedge ... \wedge C_m$ is satisfiable. First suppose that $\mathcal{P}$ is a potential solution, and let ν be the valuation defined as follows: $\nu(x_j) = \mathsf{true}$ iff there exists some $P(c_i, x_j) \in \mathcal{A}_0 \setminus \mathcal{P}$. Because $\mathcal{P}$ satisfies all wanted answers, we know that for every C_i, the ABox $\mathcal{A}_0 \setminus \mathcal{P}$ contains an assertion of the form $P(c_i, x_j)$ or $N(c_i, x_j)$. In the former case, $\nu(x_j) = \mathsf{true}$, so ν satisfies C_i. In the latter case, since $\mathcal{P}$ satisfies the unwanted answer, $N(c_i, x_j) \in \mathcal{A}_0 \setminus \mathcal{P}$ implies that $\nu(x_j) = \mathsf{false}$, so ν satisfies C_i.

Conversely, if ν is a valuation of $x_1, ..., x_n$ that satisfies the set of clauses, then $\mathcal{P} = \{P(c_i, x_j) | \nu(x_j) = \mathsf{false}\} \cup \{N(c_i, x_j) | \nu(x_j) = \mathsf{true}\}$ is a potential solution: it satisfies every $q \in \mathcal{U}$ since no x_j can have both incoming P- and N-edges in $\mathcal{A}_0 \backslash \mathcal{P}$, and every $q \in \mathcal{W}$ because every clause contains some x_j with $\nu(x_j) = \mathsf{true}$ or $\neg x_j$ with $\nu(x_j) = \mathsf{false}$, so every c_i has an outgoing P- or N-edge in $\mathcal{A}_0 \backslash \mathcal{P}$.

MCSWs: The proof is by reduction from deciding if a set of clauses of an unsatisfiable set of clauses $\{C_1, ..., C_m\}$ is a MCS, using the same QRP setting as for existence. Since the set of clauses is unsatisfiable, there does not exist a potential solution. The MCSWs correspond to the MCSes of $\{C_1, ..., C_m\}$. Indeed, a set $\{S(c_{i_1}), ..., S(c_{i_k})\}$ is a MCSW iff there exists a potential solution with $\mathcal{W}' = \mathcal{W} \backslash \{S(c_{i_1}), ..., S(c_{i_k})\}$ and for every $\mathcal{M} \subsetneq \{S(c_{i_1}), ..., S(c_{i_k})\}$ there is no potential solution with $\mathcal{W}' = \mathcal{W} \backslash \mathcal{M}$. Using the same arguments as in reduction for existence, one can show that the set of clauses $\{C_1, ..., C_m\} \backslash \{C_{i_1}, ..., C_{i_k}\}$ is satisfiable and for every $M \subsetneq \{C_{i_1}, ..., C_{i_k}\}$, $\{C_1, ..., C_m\} \backslash M$ is unsatisfiable. Indeed, a potential solution for $\mathcal{W} \backslash \{S(c_{i_1}), ..., S(c_{i_k})\}$ corresponds to a valuation that satisfies $\{C_1, ..., C_m\} \backslash \{C_{i_1}, ..., C_{i_k}\}$, and if there was a valuation satisfying $\{C_1, ..., C_m\} \backslash M$ for some $M \subsetneq \{C_{i_1}, ..., C_{i_k}\}$, there would be a potential solution for the corresponding $\mathcal{W} \backslash \mathcal{M}$. The argument in the other direction proceeds analogously.

Necessarily nonfalse: The proof is by reduction from unsatisfiability of $C_1 \wedge ... \wedge C_{m+1}$ given that $C_1 \wedge ... \wedge C_m$ is satisfiable (cf. Lemma 5.2.15). We use the same TBox $\mathcal{T}_0$ and set $\mathcal{U}$ of unwanted answers as before, together with the following slightly modified ABox and set of wanted answers:

$$\mathcal{A}_1 = \{P(c_i, x_j) | x_j \in C_i\} \cup \{N(c_i, x_j) | \neg x_j \in C_i\} \cup \{S(c_{m+1})\}$$
$$\mathcal{W}_1 = \{S(c_1), \ldots, S(c_{m+1})\}$$

We argue that the assertion $S(c_{m+1})$ is necessarily nonfalse iff $C_1 \wedge ... \wedge C_{m+1}$ is unsatisfiable. Since there exists a valuation ν that satisfies $C_1 \wedge ... \wedge C_m$, the repair plan

$$\begin{aligned}\mathcal{P} = &\{P(c_i, x_j) | \nu(x_j) = \mathsf{false}\} \cup \{N(c_i, x_j) | \nu(x_j) = \mathsf{true}\} \cup \\ &\{P(c_{m+1}, x_j), N(c_{m+1}, x_j)\} \cap \mathcal{A}_1\end{aligned}$$

is a potential solution: the wanted answer $S(c_{m+1})$ is satisfied by the assertion $S(c_{m+1})$, the other wanted answers are satisfied by outgoing P- or N-edges as in proof for existence. The set of clauses $C_1 \wedge ... \wedge C_{m+1}$ is unsatisfiable iff no potential solution contains the assertion $S(c_{m+1})$ (i.e., we are forced to keep $S(c_{m+1})$ to satisfy the wanted answers).

Necessarily false: The proof is by reduction from unsatisfiability of $C_1 \wedge ... \wedge C_{m+1}$ given that $C_1 \wedge ... \wedge C_m$ is satisfiable (cf. Lemma 5.2.15). We reuse the sets $\mathcal{U}$ and $\mathcal{W}_1$ of unwanted and wanted answers from before, and consider the following TBox and ABox:

$$\mathcal{T}_2 = \mathcal{T}_0 \cup \{E \sqsubseteq S, U \sqsubseteq \neg E\}$$
$$\mathcal{A}_2 = \{P(c_i, x_j) | x_j \in C_i\} \cup \{N(c_i, x_j) | \neg x_j \in C_i\} \cup \{E(c_{m+1}), U(c_{m+1})\}$$

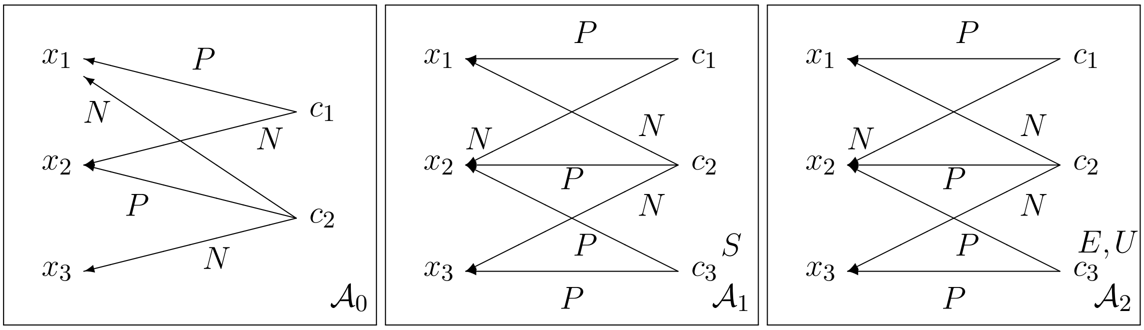
We show that $U(c_{m+1})$ is necessarily false iff $C_1 \wedge ... \wedge C_{m+1}$ is unsatisfiable. Since there exists a valuation ν that satisfies $C_1 \wedge ... \wedge C_m$, the repair plan

$$\begin{aligned}\mathcal{P} = &\{P(c_i, x_j) | \nu(x_j) = \mathsf{false}\} \cup \{N(c_i, x_j) | \nu(x_j) = \mathsf{true}\} \cup \{U(c_{m+1})\} \cup \\ &\{P(c_{m+1}, x_j), N(c_{m+1}, x_j)\} \cap \mathcal{A}_2\end{aligned}$$

is a potential solution: the wanted answer $S(c_{m+1})$ is satisfied by the assertion $E(c_{m+1})$, and the other wanted answers are satisfied by outgoing P- or N-edges as in proof for existence. The set of clauses $C_1 \wedge ... \wedge C_{m+1}$ is unsatisfiable iff every potential solution is such that $S(c_{m+1})$ is satisfied by means of the assertion $E(c_{m+1})$, so the conflicting assertion $U(c_{m+1})$ is included in the potential solution. □

Remark 5.3.6 (Minimal incompatible sets of wanted answers)**.** If we drop the truthfulness condition and assume that the user may make mistakes when inputting the wanted and unwanted answers, it may be interesting to show him the minimal sets of wanted answers that are incompatible instead of the MCSWs to help him to find its error. Indeed, the former show a "problem" that must be solved by removing one of the answers of this set, which should not be wanted, while it may be difficult to choose which MCSW to remove. However, in the case where we assume that the user cannot make mistake, the MCSWs are more appropriate,

Fig. 5.3 Reductions for hardness of problems related to potential solutions. Graphical representation of the ABox constructed from an example set of clauses $\varphi = \{C_1 = X_1 \vee \neg X_2, C_2 = \neg X_1 \vee X_2 \vee \neg X_3\}$ (or $\varphi = \{C_1 = X_1 \vee \neg X_2, C_2 = \neg X_1 \vee X_2 \vee \neg X_3\} \cup \{C_3 = x_2 \vee x_3\}$).



since in all cases the user has to look into the causes of the answers to find those which are supported only by erroneous data.

The minimal incompatible sets of wanted answers correspond to the MUSes of S w.r.t. H, so can also be computed using φ. Moreover, the reduction given for the MCSWs can be used for the minimal incompatible sets of wanted answers that correspond exactly to the MUSes of the set of clauses, so the complexity results for the MUSes transfer (in particular, recognition is BH_2-complete).

5.3.2 Algorithm for optimal deletion-only repair plans

We present an algorithm OptDeletionRepairPlan (Algorihm 5.7) for computing optimal deletion-only repair plans. Within the algorithm, we denote by $R(\mathcal{K},\mathcal{U},\mathcal{W},\mathcal{A}')$ (resp. $N_{\mathsf{f}}(\mathcal{K},\mathcal{U},\mathcal{W},\mathcal{A}')$, $N_{\neg\mathsf{f}}(\mathcal{K},\mathcal{U},\mathcal{W},\mathcal{A}')$) the set of assertions from $\mathcal{A}' \subseteq \mathcal{A}$ that are relevant (resp. necessarily false, nonfalse) for the QRP $(\mathcal{K},\mathcal{U},\mathcal{W})$ when deletions are allowed only in $\mathcal{A}'$ (the set $\mathcal{A}'$ will be used to store assertions whose truth value is not yet determined). The general idea is that the algorithm incrementally builds a set of assertions that are false according to the user. It aids the user by suggesting assertions to remove, or wanted answers that might not be satisfiable when there is no potential solution, while taking into account the knowledge the user has already provided. If there exists a potential solution, the algorithm computes the necessarily (non)false assertions (Step 3) and asks the user either to validate them or to input false and nonfalse assertions to justify why they cannot be validated (Step 23). When the necessarily (non)false assertions have been validated, the user is asked to input further true or false assertions if the current set of false assertions does not address all defects (Step 15). When a potential solution is found (Step 6), the user has to verify that each wanted answer has a cause that does not contain any false assertion. If there does not exist a potential solution at some point (Step 25), either initially or after some user inputs, the algorithm looks for an erroneous MCSW by computing all MCSWs, then showing for each of them the assertions involved in the causes of each query of the MCSW. If there is a query which has a cause without any false assertion, the MCSW under examination is not erroneous, nor are the other MCSWs that contain that query. Otherwise, the MCSW is erroneous and its queries are removed from $\mathcal{W}$, and we return to the case where a potential solution exists.

Theorem 5.3.7. *The algorithm* OptDeletionRepairPlan *always terminates, and it outputs an optimal deletion-only repair plan.*

Proof. Termination follows from the fact that every time we return to Step 2, something has either been added to $\mathcal{P}_-$ or deleted from $\mathcal{W}$, nothing is ever removed from $\mathcal{P}_-$ or added to $\mathcal{W}$, and only assertions from the original ABox $\mathcal{A}$ can be added to $\mathcal{P}_-$.

Note first the following invariants:

• The set $\mathcal{P}_-$ contains only false assertions, since every time $\mathcal{P}_-$ is modified, the assertions added have been marked as false by the user, or are conflicts of assertions that have been declared true. Hence, the output plan is validatable.

• The set $\mathcal{P}_- \cup \mathcal{A}'$ contains all assertions $\alpha \in \mathcal{A}$ such that $\mathsf{user}(\alpha) = \mathsf{false}$. Indeed, $\mathcal{A}'$ is initialized to $\mathcal{A}$, and whenever α is removed from $\mathcal{A}'$, it is either added to $\mathcal{P}_-$, or it has been shown to be nonfalse.

• The satisfiable answers (i.e. those that fulfil the conditions of Proposition 3) are never removed from $\mathcal{U}$ and $\mathcal{W}$. Indeed, $\mathcal{U}$ is never modified and $\mathcal{W}$ is modified only at Step 38, where only answers that do not fulfil the conditions of Proposition 3 are removed from $\mathcal{W}$, since all their causes contain some false assertion. It follows that if at some point $\mathcal{P}_-$ satisfies every answer in $\mathcal{U} \cup \mathcal{W}$, then $\mathcal{P}_-$ is optimal.

The algorithm can end at three different steps:

- If the algorithm ends at Step 10, then $\mathcal{P}_-$ is a potential solution for $(\mathcal{K}_0, \mathcal{U}, \mathcal{W})$. That means that for every $q \in \mathcal{U}$, $\langle \mathcal{T}, \mathcal{A} \backslash \mathcal{P}_- \rangle \not\models_{\text{brave}} q$, i.e. q is satisfied by $\mathcal{P}_-$, and for every $q \in \mathcal{W}$, $\langle \mathcal{T}, \mathcal{A} \backslash \mathcal{P}_- \rangle \models_{\text{IAR}} q$. Moreover, for every $q \in \mathcal{W}$, Step 7 ensures that there is a cause of q in $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \backslash \mathcal{P}_- \rangle$ without conflicts that contains no false assertions, so q is satisfied by $\mathcal{P}_-$. It follows that $\mathcal{P}_-$ satisfies every satisfiable answer since such answers always remain in $\mathcal{U} \cup \mathcal{W}$. The output set $\mathcal{P}_-$ is thus an optimal deletion-only repair plan.
- If the algorithm ends at Step 17, the user has been required to input some false or true assertions at Step 15 and he was not able to input anything, so the user has deleted all false assertions he knows among the relevant assertions, and thus it is not possible to improve the current repair plan further. Indeed, the set of relevant assertions contains every assertion that appear in a cause of $q \in \mathcal{U} \cup \mathcal{W}$ or in a conflict of a cause of $q \in \mathcal{W}$ and has not be declared false, true or nonfalse yet, so it is not possible to satisfy additional answers by removing further assertions that are not relevant, either because they are not involved in the problem at all, or because they are known to be nonfalse.
- If the algorithm ends at Step 38, Step 2 of the general algorithm $\mathsf{OptimalRepairPlan}_{\mathcal{U}}$ is applied: the user is asked to mark every false and true assertion in the relevant assertions, so the output is optimal since it takes into account everything the user knows. □

We illustrate OptDeletionRepairPlan running on a small example.

Algorithm 5.7 OptDeletionRepairPlan

Input: QRP ($\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle, \mathcal{U}, \mathcal{W}$)
Output: repair plan
(Note: below $\mathcal{K}$ is a macro for $(\mathcal{T}, \mathcal{A} \setminus \mathcal{P}_-)$, using the current $\mathcal{P}_-$.)

1: $\mathcal{K}_0 \leftarrow \mathcal{K}$, $\mathcal{A}' \leftarrow \mathcal{A}$, $\mathcal{P}_- \leftarrow \emptyset$
2: **if** a potential solution for $(\mathcal{K}, \mathcal{U}, \mathcal{W})$ exists in $\mathcal{A}'$ **then**
3: $\quad R \leftarrow R(\mathcal{K}, \mathcal{U}, \mathcal{W}, \mathcal{A}')$, $N_\mathsf{f} \leftarrow N_\mathsf{f}(\mathcal{K}, \mathcal{U}, \mathcal{W}, \mathcal{A}')$, $N_{\neg\mathsf{f}} \leftarrow N_{\neg\mathsf{f}}(\mathcal{K}, \mathcal{U}, \mathcal{W}, \mathcal{A}')$
4: $\quad$ **if** the user validates $\forall \alpha \in N_\mathsf{f}$, $\mathsf{user}(\alpha) = \mathsf{false}$, and $\forall \alpha \in N_{\neg\mathsf{f}}$, $\mathsf{user}(\alpha) \neq \mathsf{false}$ **then**
5: $\quad\quad \mathcal{P}_- \leftarrow \mathcal{P}_- \cup N_\mathsf{f}$, $\mathcal{A}' \leftarrow \mathcal{A}' \setminus (N_\mathsf{f} \cup N_{\neg\mathsf{f}})$
6: $\quad\quad$ **if** $\mathcal{P}_-$ is a potential solution for $(\mathcal{K}_0, \mathcal{U}, \mathcal{W})$ **then**
7: $\quad\quad\quad$ the user gives *all* false assertions $F \subseteq \bigcup_{q \in \mathcal{W}, \mathcal{C} \in \mathsf{causes}(q, \mathcal{K}), \mathsf{confl}(\mathcal{C}, \mathcal{K}) = \emptyset} \mathcal{C}$
8: $\quad\quad\quad \mathcal{P}_- \leftarrow \mathcal{P}_- \cup F$
9: $\quad\quad\quad$ **if** $\mathcal{P}_-$ is still a potential solution **then**
10: $\quad\quad\quad\quad$ output $\mathcal{P}_-$
11: $\quad\quad\quad$ **else**
12: $\quad\quad\quad\quad \mathcal{A}' \leftarrow \mathcal{A}' \setminus \mathcal{P}_-$, go to 2
13: $\quad\quad\quad$ **end if**
14: $\quad\quad$ **else**
15: $\quad\quad\quad$ user selects *some* $F, T \subseteq R \setminus (N_\mathsf{f} \cup N_{\neg\mathsf{f}})$
16: $\quad\quad\quad$ **if** $F = T = \emptyset$ **then** // nothing left to input
17: $\quad\quad\quad\quad$ return $\mathcal{P}_-$
18: $\quad\quad\quad$ **else**
19: $\quad\quad\quad\quad \mathcal{P}_- \leftarrow \mathcal{P}_- \cup F \cup \mathsf{confl}(T,)$, $\mathcal{A}' \leftarrow \mathcal{A}' \setminus (\mathcal{P}_- \cup T)$, go to 2
20: $\quad\quad\quad$ **end if**
21: $\quad\quad$ **end if**
22: $\quad$ **else**
23: $\quad\quad$ user gives $F \subseteq \{\alpha \in N_{\neg\mathsf{f}} \mid \mathsf{user}(\alpha) = \mathsf{false}\}$ and $NF \subseteq \{\alpha \in N_\mathsf{f} \mid \mathsf{user}(\alpha) \neq \mathsf{false}\}$ with $F \cup NF \neq \emptyset$, $\mathcal{P}_- \leftarrow \mathcal{P}_- \cup F$, $\mathcal{A}' \leftarrow \mathcal{A}' \setminus (\mathcal{P}_- \cup NF)$
24: $\quad$ **end if**
25: **else** // search for an erroneous MCSW
26: $\quad \mathcal{M} \leftarrow MCSWs(\mathcal{K}, \mathcal{U}, \mathcal{W}, \mathcal{A}')$ ordered by size
27: $\quad$ **while** erroneous MCSW not found and $\mathcal{M} \neq \emptyset$ **do**
28: $\quad\quad M \leftarrow$ first$(\mathcal{M})$
29: $\quad\quad$ **for all** $q \in M$ **do**
30: $\quad\quad\quad$ the user selects $F, T \subseteq \bigcup_{\mathcal{C} \in \mathsf{causes}(q, \mathcal{K})} \mathcal{C}$
31: $\quad\quad\quad \mathcal{P}_- \leftarrow \mathcal{P}_- \cup F \cup \mathsf{confl}(T,)$, $\mathcal{A}' \leftarrow \mathcal{A}' \setminus (\mathcal{P}_- \cup T)$
32: $\quad\quad\quad$ **if** a cause for q contains no false assertion **then**
33: $\quad\quad\quad\quad \mathcal{M} \leftarrow \mathcal{M} \setminus \{M' \in \mathcal{M} \mid q \in M'\}$, go to 27
34: $\quad\quad\quad$ **end if**
35: $\quad\quad$ **end for**
36: $\quad\quad$ MCSW found: $\mathcal{W} \leftarrow \mathcal{W} \setminus M$ and go to 3
37: $\quad$ **end while**
38: $\quad$ No MCSW found: do Step 2 of $\mathsf{OptimalRepairPlan}_\mathcal{U}$, output $\mathcal{P}_-$
39: **end if**

Example 5.3.8. Suppose that we have the following QRP ($\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$,$\mathcal{U}$,$\mathcal{W}$).

$$\begin{aligned}
\mathcal{T} =& \{\mathsf{AProf} \sqsubseteq \mathsf{Prof}, \mathsf{Prof} \sqsubseteq \mathsf{Employee}, \mathsf{Postdoc} \sqsubseteq \mathsf{Employee}, \exists \mathsf{MemberOf} \sqsubseteq \mathsf{Employee}\} \\
\mathcal{A} =& \{\mathsf{Teach}(a,b), \mathsf{GradCourse}(b), \mathsf{AProf}(a), \mathsf{Postdoc}(a), \mathsf{MemberOf}(a,c), \\
& \mathsf{PublicationAuthor}(a,d)\} \\
\mathcal{U} =& \{q_1 = \exists x \mathsf{AProf}(x) \wedge \mathsf{Teach}(x,b), \\
& q_2 = \exists xy \mathsf{Teach}(x,y) \wedge \mathsf{GradCourse}(y) \wedge \mathsf{Postdoc}(x)\} \\
\mathcal{W} =& \{q_3 = \exists x \mathsf{Teach}(x,b), q_4 = \mathsf{Prof}(a), q_5 = \exists x \mathsf{Employee}(a) \wedge \mathsf{PublicationAuthor}(a,x)\}
\end{aligned}$$

The first step is to check if there exists a potential solution for ($\mathcal{K}$,$\mathcal{U}$,$\mathcal{W}$) in $\mathcal{A}' = \mathcal{A}$. This is not the case because q_1, q_3, and q_4 cannot be satisfied together because q_3 and q_4 have only one cause each ($\{\mathsf{Teach}(a,b)\}$ and $\{\mathsf{AProf}(a)\}$ respectively), whose union forms a cause for the unwanted answer q_1. Therefore, we jump to Step 25 and compute the MCSWs. There are two MCSWs: $\{q_3\}$ and $\{q_4\}$. We first display the assertions of the causes for q_3 (i.e. $\mathsf{Teach}(a,b)$) and ask the user if he knows some true or false assertions. Suppose that $\mathsf{user}(\mathsf{Teach}(a,b)) = \mathsf{unknown}$. Since a cause for q_3 contains no false assertion, $\{q_3\}$ is not an erroneous MCSW and we examine the next one. We then present the assertions of the causes for q_4 ($\mathsf{AProf}(a)$). Suppose that the user indicates that $\mathsf{user}(\mathsf{AProf}(a)) = \mathsf{false}$. Since $\mathsf{AProf}(a)$ is false, it is added to $\mathcal{P}_-$ and removed from $\mathcal{A}'$. At this point, q_4 has no cause in $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \backslash \mathcal{P}_- \rangle$, so $\{q_4\}$ is an erroneous MCSW: q_4 is removed from $\mathcal{W}$ and we jump to Step 3.

We know that there exists some potential solution and compute the relevant, necessarily false and nonfalse assertions: $N_{\mathsf{f}} = \emptyset$, $N_{\neg\mathsf{f}} = \{\mathsf{Teach}(a,b), \mathsf{PublicationAuthor}(a,d)\}$. Suppose that the user validates that the two assertions of $N_{\neg\mathsf{f}}$ are nonfalse. Then $\mathsf{Teach}(a,b)$ and $\mathsf{PublicationAuthor}(a,d)$ are removed from $\mathcal{A}'$.

$\mathcal{P}_- = \{\mathsf{AProf}(a)\}$ is not a potential solution (since q_2 is not satisfied), so the relevant assertions ($\mathsf{GradCourse}(b)$, $\mathsf{Postdoc}(a)$ and $\mathsf{MemberOf}(a,c)$) are displayed. The user indicates that $\mathsf{Postdoc}(a)$ is false, so $\mathsf{Postdoc}(a)$ is added to $\mathcal{P}_-$ and removed from $\mathcal{A}'$. We then go to Step 2.

A potential solution exists (actually $\mathcal{P}_-$ is a potential solution for ($\mathcal{K}_0$,$\mathcal{U}$,$\mathcal{W}$), so $\emptyset$ is a potential solution for ($\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \backslash \mathcal{P}_- \rangle$,$\mathcal{U}$,$\mathcal{W}$)). The user validates the necessarily (non)false assertions (here all remaining assertions are necessarily nonfalse), and the assertions involved in some IAR causes of the wanted answers are displayed (here all assertions of $\mathcal{A} \backslash \mathcal{P}_-$). Suppose that the user does not find anything false, then we have found an optimal deletion-only repair plan: $\mathcal{P}_- = \{\mathsf{AProf}(a), \mathsf{Postdoc}(a)\}$. $\triangleleft$

5.3.3 Improvements to the algorithm

To avoid overwhelming the user with relevant assertions at Step 15, it is desirable to reduce the number of assertions presented at a time. This leads us to propose two improvements to the basic algorithm.

First, we can divide QRPs into *independent subproblems*. Two answers are considered dependent if their causes (and conflicts in the case of wanted answers) share some assertion.

Independent sets of answers do not interact, so they can be handled separately. Algorithm 5.8 computes independent subproblems of a QRP by constructing the independent sets of answers. For each wanted or unwanted answer q_0, the algorithm first builds a set that contains only this answer. Then for each independent set p it has already built, it checks whether some query of p is dependent of q_0. This is the case if some query of p has causes (or conflicts if it is a wanted query), that share assertions with the causes of q_0 (or its conflicts if q_0 is wanted). If so, it aggregates p with the set of q_0. At the end of the algorithm, each query appears in exactly one set, and every pair of dependent answers are in the same set.

Algorithm 5.8 IndependentSubproblems

Input: QRP $(\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle, \mathcal{U}, \mathcal{W})$
Output: independent QRPs $\{(\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle, \mathcal{U}_i, \mathcal{W}_i)\}$

1: $P \leftarrow \emptyset$
2: **for all** $q_0 \in \mathcal{U} \cup \mathcal{W}$ **do**
3: $\quad p_{q_0} \leftarrow \{q_0\}$
4: $\quad$ **if** $q_0 \in \mathcal{W}$ **then**
5: $\quad\quad B_0 \leftarrow \bigcup_{\mathcal{C} \in \mathsf{causes}(q_0, \mathcal{K})} (\mathcal{C} \cup \mathsf{confl}(\mathcal{C}, \mathcal{K}))$
6: $\quad$ **else**
7: $\quad\quad B_0 \leftarrow \bigcup_{\mathcal{C} \in \mathsf{causes}(q_0, \mathcal{K})} \mathcal{C}$
8: $\quad$ **end if**
9: $\quad$ **for all** $p \in P$ **do**
10: $\quad\quad B_p \leftarrow \bigcup_{q \in p \cap \mathcal{W}, \mathcal{C} \in \mathsf{causes}(q, \mathcal{K})} (\mathcal{C} \cup \mathsf{confl}(\mathcal{C}, \mathcal{K})) \cup \bigcup_{q \in p \cap \mathcal{U}, \mathcal{C} \in \mathsf{causes}(q, \mathcal{K})} \mathcal{C}$
11: $\quad\quad$ **if** $B_p \cap B_0 \neq \emptyset$ **then**
12: $\quad\quad\quad P \leftarrow P \setminus \{p\}$
13: $\quad\quad\quad p_{q_0} \leftarrow p_{q_0} \cup p$
14: $\quad\quad$ **end if**
15: $\quad$ **end for**
16: $\quad P \leftarrow P \cup \{p_{q_0}\}$
17: **end for**
18: Output $\{(\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle, \mathcal{U} \cap p_i, \mathcal{W} \cap p_i) \mid p_i \in P\}$

Second, at Step 15, the assertions can be presented in small batches. When axiom (in)validation can be partially automatized, ranking axioms by their potential *impact* reduces the effort of manual revision [Meilicke *et al.* 2008, Nikitina *et al.* 2012]. In our setting, we believe that validating sets of necessarily (non)false assertions requires less effort than hunting for false assertions among all relevant assertions, leading us to propose a similar notion of impact to rank assertions to be examined. Indeed, deleting or keeping an assertion may force us to delete or keep other assertions to get a potential solution. Relevant assertions can be sorted using two scores that express the impact of being declared false or true. For the impact of an assertion α being false, we use the number of assertions that becomes necessarily (non)false if α is deleted. The impact of α being true also takes into account the fact that the conflicts of α can be marked as false: we consider the number of assertions that are in conflict with α or become necessarily (non)false when we disallow α's removal. We can rank assertions by the minimum of the two scores, using their sum to break ties. More formally, the scores are given by the following formulas:

$$\mathsf{impact}_{\mathsf{false}}(\alpha) = |N_{\mathsf{f}}(\alpha = \mathsf{false}) \cup N_{\neg\mathsf{f}}(\alpha = \mathsf{false})|$$
$$\mathsf{impact}_{\mathsf{true}}(\alpha) = |N_{\mathsf{f}}(\alpha = \mathsf{true}) \cup N_{\neg\mathsf{f}}(\alpha = \mathsf{true})| + |(\mathsf{confl}(\{\alpha\},\mathcal{K}) \cap R) \setminus N_{\mathsf{f}}(\alpha = \mathsf{true})|$$

with

$$R = R(\langle \mathcal{T},\mathcal{A}\rangle,\mathcal{U},\mathcal{W},\mathcal{A}'),$$
$$N_{(\neg)\mathsf{f}}(\alpha = \mathsf{false}) = N_{(\neg)\mathsf{f}}(\langle \mathcal{T},\mathcal{A}\setminus\{\alpha\}\rangle,\mathcal{U},\mathcal{W},\mathcal{A}'\setminus\{\alpha\}),$$
$$N_{(\neg)\mathsf{f}}(\alpha = \mathsf{true}) = N_{(\neg)\mathsf{f}}(\langle \mathcal{T},\mathcal{A}\rangle,\mathcal{U},\mathcal{W},\mathcal{A}'\setminus\{\alpha\}).$$

Example 5.3.9 (Example 5.3.8 cont'd)**.** At Step 3, there are three assertions to sort:

$$\mathsf{GradCourse}(b) \qquad \mathsf{Postdoc}(a) \qquad \mathsf{MemberOf}(a,c)$$

Their impacts are computed as follows:

$$\mathsf{impact}_{\mathsf{false}}(\mathsf{GradCourse}(b)) = 0$$
$$\mathsf{impact}_{\mathsf{true}}(\mathsf{GradCourse}(b)) = |\{\mathsf{Postdoc}(a)\}| = 1$$

Indeed, if $\mathsf{GradCourse}(b)$ is true, then $\mathsf{Postdoc}(a)$ is necessarily false. It follows that $\min(\mathsf{impact}_{\mathsf{false}}(\mathsf{GradCourse}(b)), \mathsf{impact}_{\mathsf{true}}(\mathsf{GradCourse}(b))) = 0$ and $\mathsf{impact}_{\mathsf{false}}(\mathsf{GradCourse}(b)) + \mathsf{impact}_{\mathsf{true}}(\mathsf{GradCourse}(b)) = 1$.

$$\mathsf{impact}_{\mathsf{false}}(\mathsf{Postdoc}(a)) = |\{\mathsf{MemberOf}(a,c)\}| = 1$$
$$\mathsf{impact}_{\mathsf{true}}(\mathsf{Postdoc}(a)) = |\{\mathsf{GradCourse}(b)\}| = 1$$

Indeed, if $\mathsf{Postdoc}(a)$ is false, then $\mathsf{MemberOf}(a,c)$ is necessarily nonfalse and if $\mathsf{Postdoc}(a)$ is true, then $\mathsf{GradCourse}(b)$ is necessarily false. It follows that $\min(\mathsf{impact}_{\mathsf{false}}(\mathsf{Postdoc}(a)), \mathsf{impact}_{\mathsf{true}}(\mathsf{Postdoc}(a))) = 1$ and $\mathsf{impact}_{\mathsf{false}}(\mathsf{Postdoc}(a)) + \mathsf{impact}_{\mathsf{true}}(\mathsf{Postdoc}(a)) = 2$.

$$\mathsf{impact}_{\mathsf{false}}(\mathsf{MemberOf}(a,c)) = |\{\mathsf{Postdoc}(a)\}| = 1$$
$$\mathsf{impact}_{\mathsf{true}}(\mathsf{MemberOf}(a,c)) = 0$$

Indeed, if $\mathsf{MemberOf}(a,c)$ is false, $\mathsf{Postdoc}(a)$ is necessarily nonfalse. We thus have $\min(\mathsf{impact}_{\mathsf{false}}(\mathsf{MemberOf}(a,c)), \mathsf{impact}_{\mathsf{true}}(\mathsf{MemberOf}(a,c))) = 0$ and $\mathsf{impact}_{\mathsf{false}}(\mathsf{MemberOf}(a,c)) + \mathsf{impact}_{\mathsf{true}}(\mathsf{MemberOf}(a,c)) = 1$.

Based upon the impact scores, $\mathsf{Postdoc}(a)$ will be presented first. ◁

5.4 Considering the AR semantics for wanted answers

While in the preceding chapters we studied three semantics, AR, IAR and brave, we defined the query-driven repairing problem using only the brave and IAR semantics. In this section, we investigate the use of AR semantics in this context. We still call potential solutions the repair plans that achieve our objectives in terms of query (non) entailments. More formally, a *potential solution for AR (resp. for IAR)* is a subset of the ABox whose removal makes every unwanted answer not hold under brave semantics, and every wanted answer hold under AR (resp. IAR) semantics.

One could think that requiring that wanted answers hold under AR semantics instead of IAR semantics would help to find potential solutions or yield more necessarily nonfalse assertions. The following proposition shows that this is not true because a potential solution for AR can always be transformed into a potential solution for IAR.

Proposition 5.4.1. *Let $(\langle\mathcal{T},\mathcal{A}\rangle,\mathcal{U},\mathcal{W})$ be a QRP. If there does not exist a potential solution for IAR, then there does not exist a potential solution for AR.*

Proof. If $(\mathcal{P}_-^{AR},\mathcal{P}_+^{AR})$ is a repair plan for $(\langle\mathcal{T},\mathcal{A}\rangle,\mathcal{U},\mathcal{W})$ that makes the wanted answers hold under AR semantics and unwanted answers not hold under brave semantics, let $\mathcal{R}$ be a repair of $(\mathcal{A}\backslash\mathcal{P}_-^{AR})\cup\mathcal{P}_+^{AR}$. Then $\mathcal{P}^{IAR}=(\mathcal{A}\backslash\mathcal{R},\mathcal{P}_+^{AR}\cap\mathcal{R})$ is a repair plan that makes the wanted answers hold under IAR semantics and unwanted answers not hold under brave semantics. Indeed, $(\mathcal{A}\backslash\mathcal{P}_-^{IAR})\cup\mathcal{P}_+^{IAR}=\mathcal{R}$ and every $q\in\mathcal{W}$ has a cause in $\mathcal{R}$, which has no conflicts by consistency of $\mathcal{R}$. □

Futhermore, deciding if a set of assertions is a potential solution for AR is intractable, while it is tractable for IAR.

Proposition 5.4.2. *For complexity w.r.t. $|\mathcal{A}|$, $|\mathcal{U}|$ and $|\mathcal{W}|$, deciding if $(\mathcal{P}_-,\mathcal{P}_+)$ is a potential solution for AR is co*NP*-complete.*

Proof. We can show that $\mathcal{P}$ is *not* a potential solution for AR as follows: decide in P if all unwanted answers are satisfied, and if it is the case, guess a wanted answer q and a repair $\mathcal{R}$ of $\langle\mathcal{T},(\mathcal{A}\backslash\mathcal{P}_-)\cup\mathcal{P}_+\rangle$ such that $\langle\mathcal{T},\mathcal{R}\rangle\not\models q$.

Hardness is by reduction from AR query answering: $\langle\mathcal{T},\mathcal{A}\rangle\models_{\text{AR}} q$ iff $(\emptyset,\emptyset)$ is a potential solution for AR for the QRP $(\langle\mathcal{T},\mathcal{A}\rangle,\emptyset,\{q\})$. □

However, the following example shows that it may be useful to accept that some wanted answers are entailed under AR semantics rather than IAR, because even if using AR semantics has no impact on the existence of potential solutions, it may have one on existence of validatable repair plans that satisfy all wanted answers.

Example 5.4.3 (Example 5.1.1 cont'd)**.** Consider the following QRP:

$$\begin{aligned}\mathcal{K}=&(\mathcal{T}_{ex},\{\mathsf{FProf}(a),\mathsf{AProf}(a),\mathsf{Postdoc}(a)\})\\ \mathcal{W}=&\{\mathsf{Prof}(a)\},\mathcal{U}=\emptyset\end{aligned}$$

Suppose that $\mathsf{user}(\mathsf{FProf}(a)) = \mathsf{unknown}$, $\mathsf{user}(\mathsf{AProf}(a)) = \mathsf{unknown}$ and $\mathsf{user}(\mathsf{Postdoc}(a)) = \mathsf{false}$. There are two deletion-only potential solutions: $\{\mathsf{FProf}(a), \mathsf{Postdoc}(a)\}$ and $\{\mathsf{AProf}(a), \mathsf{Postdoc}(a)\}$, but none of them is validatable. However, deleting $\mathsf{Postdoc}(a)$ is a validatable deletion-only repair-plan that makes $\mathsf{Prof}(a)$ hold under AR semantics.

Even in the case where additions are allowed, we would probably prefer to simply delete $\mathsf{Postdoc}(a)$ and keep the information than a is either a full or an assistant professor, rather than also adding $\mathsf{Prof}(a)$ to create an IAR cause which is somehow redundant with the knowledge already present in the base. Moreover, it is possible that in some settings it would be forbidden to add assertions using the general predicate Prof, for instance if a professor has to be registered with his level of seniority. $\triangleleft$

In this section, we therefore study the impact of using AR semantics to define the satisfaction of wanted answers. We refine the notions of satisfaction (resp. satisfiability, satisfiability w.r.t. a repair plan) of a wanted answer into satisfaction (resp. satisfiability, satisfiability w.r.t. a repair plan) for IAR and for AR. We say that q is *satisfied by $\mathcal{K}$ for S (S $\in \{$IAR,AR$\}$)* if there exists an explanation for $\mathcal{K} \models_{\mathrm{S}} q$ that does not contain any false assertions. In the case of AR semantics, this means that there exist $\mathcal{C}_1, \ldots, \mathcal{C}_k \in \mathsf{causes}(q, \mathcal{K})$ such that every repair of $\mathcal{K}$ contains some $\mathcal{C}_i$ and there is no $\alpha \in \bigcup_{i=1}^{k} \mathcal{C}_i$ such that $\mathsf{user}(\alpha) = \mathsf{false}$. We use $\mathcal{S}_{\mathcal{W}}^{\mathrm{AR}}(\mathcal{P})$ (resp. $\mathcal{S}_{\mathcal{W}}^{\mathrm{IAR}}(\mathcal{P})$) to denote the set of answers satisfied for AR (resp. for IAR) by a repair plan $\mathcal{P}$. We redefine the preorder $\preceq_{\mathcal{W}}$ to take both semantics into account: $\mathcal{P} \prec_{\mathcal{W}} \mathcal{P}'$ iff:

- $\mathcal{S}_{\mathcal{W}}^{AR}(\mathcal{P}) \subset \mathcal{S}_{\mathcal{W}}^{AR}(\mathcal{P}')$ and $\mathcal{S}_{\mathcal{W}}^{IAR}(\mathcal{P}) \subseteq \mathcal{S}_{\mathcal{W}}^{IAR}(\mathcal{P}')$, or
- $\mathcal{S}_{\mathcal{W}}^{AR}(\mathcal{P}) = \mathcal{S}_{\mathcal{W}}^{AR}(\mathcal{P}')$ and $\mathcal{S}_{\mathcal{W}}^{IAR}(\mathcal{P}) \subset \mathcal{S}_{\mathcal{W}}^{IAR}(\mathcal{P}')$.

Remark 5.4.4. When we do not want to impose IAR semantics rather than AR semantics to avoid adding assertions, we can also define a weak preorder: $\mathcal{P} \prec_{\mathcal{W}}^{weak} \mathcal{P}'$ iff:

- $\mathcal{S}_{\mathcal{W}}^{AR}(\mathcal{P}) \subset \mathcal{S}_{\mathcal{W}}^{AR}(\mathcal{P}')$ and $\mathcal{S}_{\mathcal{W}}^{IAR}(\mathcal{P}) \subseteq \mathcal{S}_{\mathcal{W}}^{IAR}(\mathcal{P}')$, or
- $\mathcal{S}_{\mathcal{W}}^{AR}(\mathcal{P}) = \mathcal{S}_{\mathcal{W}}^{AR}(\mathcal{P}')$ and $\mathcal{S}_{\mathcal{W}}^{IAR}(\mathcal{P}) \subset \mathcal{S}_{\mathcal{W}}^{IAR}(\mathcal{P}')$ and $\mathcal{P}'_+ = \mathcal{P}_+$.

5.4.1 Characterization and complexity of optimal repair plans

We recall that for the problems related to optimal repair plans, we measure complexity w.r.t. $|\mathcal{A}|$, $|\mathcal{U}|$, $|\mathcal{W}|$, and $|True_{\mathsf{user}}^{rel}|$, and assume that the sizes of the queries and the TBox are bounded. In particular, this means that the causes of the queries can be computed in polynomial time.

First note that Lemma 5.2.5 is still true: if $\mathcal{P}'$ is validatable and such that $\mathcal{P}_- \subseteq \mathcal{P}'_-$ and $\mathcal{P}_+ \subseteq \mathcal{P}'_+$, then $\mathcal{S}_{\mathcal{W}}^{AR}(\mathcal{P}_-, \mathcal{P}_+) \subseteq \mathcal{S}_{\mathcal{W}}^{AR}(\mathcal{P}'_-, \mathcal{P}'_+)$ and $\mathcal{S}_{\mathcal{W}}^{IAR}(\mathcal{P}_-, \mathcal{P}_+) \subseteq \mathcal{S}_{\mathcal{W}}^{IAR}(\mathcal{P}'_-, \mathcal{P}'_+)$, with the same argument: extending the repair plan in a validatable fashion preserves the nonfalse causes of wanted answers, and cannot add assertions which conflict them. It follows that the characterizations of optimal repair plans in terms of satisfiability of answers with the redefined $\preceq_{\mathcal{W}}$ are similar to those with the initial $\preceq_{\mathcal{W}}$, since the proof of Proposition 5.2.7

uses only this lemma and the definitions of the different notions of optimality, satisfaction, satisfiability and satisfiability w.r.t. a repair plan are not modified.

Proposition 5.4.5. *A validatable repair plan $\mathcal{P}$ is:*

- *globally $\preceq_{\mathcal{W}}$-optimal iff it is locally $\preceq_{\mathcal{W}}$-optimal iff it satisfies for AR every $q \in \mathcal{W}$ satisfiable for AR, and for IAR every $q \in \mathcal{W}$ satisfiable for IAR.*
- *locally $\preceq_{\mathcal{U},\mathcal{W}}$-optimal iff it is locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal iff it satisfies every $q \in \mathcal{U}$ that is satisfiable w.r.t. $\mathcal{P}$, satisfies for AR every $q \in \mathcal{W}$ that is satisfiable for AR w.r.t. $\mathcal{P}$, and satisfies for IAR every $q \in \mathcal{W}$ that is satisfiable for IAR w.r.t. $\mathcal{P}$.*
- *locally $\preceq_{\mathcal{W},\mathcal{U}}$-optimal iff it satisfies for AR every $q \in \mathcal{W}$ that is satisfiable for AR, satisfies for IAR every $q \in \mathcal{W}$ that is satisfiable for IAR, and satisfies every $q \in \mathcal{U}$ that is satisfiable w.r.t. $\mathcal{P}$.*

As for IAR, we can give the characterizations of answers that are satisfiable or satisfiable w.r.t. a repair plan for AR.

Lemma 5.4.6. *An answer $q \in \mathcal{W}$ is satisfiable for AR iff there exists some $\mathcal{T}$-consistent sets of assertions $\mathcal{C}_1, ..., \mathcal{C}_k$ such that:*

- *for every $1 \leq i \leq k$ $\langle \mathcal{T}, \mathcal{C}_i \rangle \models q$,*
- *for every $\alpha \in \bigcup_{i=1}^{k} \mathcal{C}_i$, either $\mathsf{user}(\alpha) = \mathsf{true}$, or $\mathsf{user}(\alpha) = \mathsf{unknown}$ and $\alpha \in \mathcal{A}$, and*
- *for every consistent $\mathcal{B} \subseteq \mathcal{A}$ such that for every $1 \leq i \leq k$, $\mathcal{B} \cup \mathcal{C}_i$ is inconsistent, there exists $\beta \in \mathcal{B}$ such that $\mathsf{user}(\beta) = \mathsf{false}$.*

(We will call $\mathcal{C}_1, ..., \mathcal{C}_k$ a witness *for the satisfiability of q for AR.)*

Proof. If $q \in \mathcal{W}$ is satisfiable for AR, then there exists a validatable repair plan $(\mathcal{P}_-, \mathcal{P}_+)$ such that $(\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+$ contains an explanation $\mathcal{C}_1, ..., \mathcal{C}_k$ for $\langle \mathcal{T}, (\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+ \rangle \models_{\text{AR}} q$ that does not contain any false assertions. It follows that $\mathcal{C}_1, ..., \mathcal{C}_k$ are consistent sets of assertions such that $\langle \mathcal{T}, \mathcal{C}_i \rangle \models q$, and for every $\alpha \in \bigcup_{i=1}^{k} \mathcal{C}_i$, either $\alpha \in \mathcal{P}_+$ and $\mathsf{user}(\alpha) = \mathsf{true}$, or $\alpha \in \mathcal{A}$ and $\mathsf{user}(\alpha) = \mathsf{true}$ or $\mathsf{user}(\alpha) = \mathsf{unknown}$. Moreover, for every consistent $\mathcal{B} \subseteq \mathcal{A}$ such that $\mathcal{B} \cup \mathcal{C}_i$ is inconsistent for every $1 \leq i \leq k$, $\mathcal{B} \cap \mathcal{P}_- \neq \emptyset$. Otherwise $\mathcal{B}$ would be a consistent subset of $(\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+$ contradicting every $\mathcal{C}_i$, and $\mathcal{C}_1, ..., \mathcal{C}_k$ would not be an explanation for $\langle \mathcal{T}, (\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+ \rangle \models_{\text{AR}} q$. Since $\mathcal{P}$ is validatable, it follows that there exists $\beta \in \mathcal{B}$ such that $\mathsf{user}(\beta) = \mathsf{false}$.

In the other direction, if q and $\mathcal{C}_1, ..., \mathcal{C}_k$ satisfy the conditions of the lemma statement, then the repair plan $\mathcal{P}$ defined by:

$$(\{\beta \in \mathcal{A} \mid \exists \mathcal{B} \subseteq \mathcal{A} \text{ such that } \mathcal{B} \text{ is consistent and } \forall \mathcal{C}_i \in \{\mathcal{C}_1, ..., \mathcal{C}_k\}, \mathcal{B} \cup \mathcal{C}_i \text{ is inconsistent,}$$
$$\beta \in \mathcal{B}, \mathsf{user}(\beta) = \mathsf{false}\}, \{\alpha \in \bigcup_{i=1}^{k} \mathcal{C}_i \setminus \mathcal{A} \mid \mathsf{user}(\alpha) = \mathsf{true}\})$$

is a validatable repair plan that satisfies q for AR. Indeed, there is no consistent $\mathcal{B} \subseteq (\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+$ that contradicts $\mathcal{C}_1, ..., \mathcal{C}_k$: since $\mathcal{P}_+$ contains only true assertions, whose conflicts are false, and the $\mathcal{C}_i$ contains no false assertions, such a $\mathcal{B}$ would be included in $\mathcal{A}$, so would contain some false assertion β, which belongs to $\mathcal{P}_-$ by construction. Therefore every repair of $(\mathcal{A}\backslash\mathcal{P}_-)\cup\mathcal{P}_+$ contains some $\mathcal{C}_i$ that contains some cause for q without any false assertion, and q is satisfied for AR. □

Lemma 5.4.7. *Let $(\mathcal{P}_-,\mathcal{P}_+)$ be a validatable repair plan for the KB $\langle\mathcal{T},\mathcal{A}\rangle$. Then an answer $q\in\mathcal{W}$ is satisfiable for AR w.r.t. $(\mathcal{P}_-,\mathcal{P}_+)$ iff q is satisfiable for AR for the KB $\langle\mathcal{T},\mathcal{A}\rangle$ with a witness $\mathcal{C}_1,...,\mathcal{C}_k$ such that every $q'\in\mathcal{S}_{\mathcal{U}}(\mathcal{P}_-,\mathcal{P}_+)$ is satisfiable for the KB $\langle\mathcal{T},\mathcal{A}\cup\mathcal{P}_+\cup\bigcup_{i=1}^{k}\mathcal{C}_i\rangle$.*

Proof. If $q\in\mathcal{W}$ is satisfiable for AR w.r.t. $(\mathcal{P}_-,\mathcal{P}_+)$, then there exists a validatable repair plan $(\mathcal{P}'_-,\mathcal{P}'_+)$ such that $\mathcal{P}_-\subseteq\mathcal{P}'_-$ and $\mathcal{P}_+\subseteq\mathcal{P}'_+$ which satisfies q for AR and all answers in $\mathcal{S}(\mathcal{P}_-,\mathcal{P}_+)$. As q is satisfied for AR by $(\mathcal{P}'_-,\mathcal{P}'_+)$, the ABox $(\mathcal{A}\backslash\mathcal{P}'_-)\cup\mathcal{P}'_+$ contains an explanation $\mathcal{C}_1,...,\mathcal{C}_k$ for $\langle\mathcal{T},(\mathcal{A}\backslash\mathcal{P}'_-)\cup\mathcal{P}'_+\rangle\models_{\text{AR}} q$ that does not contain any false assertion. This means that q is satisfiable for $\langle\mathcal{T},\mathcal{A}\rangle$. We then use exactly the same arguments as in proof of Lemma 5.2.12 to show that every $q'\in\mathcal{S}_{\mathcal{U}}(\mathcal{P}_-,\mathcal{P}_+)$ is satisfiable for the KB $\langle\mathcal{T},\mathcal{A}\cup\mathcal{P}_+\cup\bigcup_{i=1}^{k}\mathcal{C}_i\rangle$.

In the other direction, suppose that $q\in\mathcal{W}$ is satisfiable for AR for the KB $\langle\mathcal{T},\mathcal{A}\rangle$ with a witness $\mathcal{C}_1,...,\mathcal{C}_k$ such that every $q'\in\mathcal{S}_{\mathcal{U}}(\mathcal{P}_-,\mathcal{P}_+)$ is satisfiable for the KB $\langle\mathcal{T},\mathcal{A}\cup\mathcal{P}_+\cup\bigcup_{i=1}^{k}\mathcal{C}_i\rangle$. We can show as in Lemma 5.2.12 that the repair plan $(\mathcal{P}'_-,\mathcal{P}'_+)$ where

$$\begin{aligned}\mathcal{P}'_- =&\mathcal{P}_-\cup\{\beta\in\mathcal{A}\mid\exists\mathcal{B}\subseteq\mathcal{A}\text{ such that }\mathcal{B}\text{ is consistent and }\forall\mathcal{C}_i\in\{\mathcal{C}_1,...,\mathcal{C}_k\},\\ &\quad\mathcal{B}\cup\mathcal{C}_i\text{ is inconsistent},\beta\in\mathcal{B},\mathsf{user}(\beta)=\mathsf{false}\}\cup\\ &\quad\{\gamma\mid q'\in\mathcal{U},\mathcal{C}\in\mathsf{causes}(q',\langle\mathcal{T},(\mathcal{A}\backslash\mathcal{P}_-)\cup\mathcal{P}_+\cup\bigcup_{i=1}^{k}\mathcal{C}_i\rangle),\gamma\in\mathcal{C},\mathsf{user}(\gamma)=\mathsf{false}\}\\ \mathcal{P}'_+ =&\mathcal{P}_+\cup\{\alpha\in\bigcup_{i=1}^{k}\mathcal{C}_i\setminus\mathcal{A}\mid\mathsf{user}(\alpha)=\mathsf{true}\}\end{aligned}$$

is a validatable repair plan that extends $\mathcal{P}$ and satisfies q for AR and all answers in $\mathcal{S}(\mathcal{P}_-,\mathcal{P}_+)$. □

In contrast to the characterizations of answers satisfiable or satisfiable w.r.t. a repair plan for IAR, these characterizations cannot be verified in P, because the size of the witness is not bounded. However, they allow us to show the following result for instance queries.

Lemma 5.4.8. *For instance queries, minimal witnesses of satisfiability for AR are either a single cause $\{\alpha\}$ with $\mathsf{user}(\alpha)=\mathsf{true}$, or a set of causes $\{\alpha_1\},...,\{\alpha_k\}$ ($k>1$) such that $\mathsf{user}(\alpha_i)=\mathsf{unknown}$ and $\alpha_i\in\mathcal{A}$ for every α_i. It follows that if an instance query q is not satisfiable for IAR w.r.t. a repair plan $\mathcal{P}$, then q is satisfiable for AR w.r.t. $\mathcal{P}$ iff q is satisfiable for AR with a witness included in $\mathcal{A}$.*

Proof. If we need to add a cause $\{\alpha\}$ with $\mathsf{user}(\alpha)=\mathsf{true}$ to satisfy an instance query q, since all its conflicts are false, removing the conflicts of α satisfies q for IAR and a minimal witness for AR satisfiability of q containing $\{\alpha\}$ does not contain any other cause.

If q is not satisfiable for IAR w.r.t. $\mathcal{P}$, then q has no true cause $\{\alpha\}$ such that every $q' \in \mathcal{S}_{\mathcal{U}}(\mathcal{P})$ is satisfiable in $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+ \cup \{\alpha\}\rangle$. If q is satisfiable for AR w.r.t. $\mathcal{P}$, its witness for AR satisfiability w.r.t. $\mathcal{P}$ $\{\alpha_1\}, ..., \{\alpha_k\}$ is then such that $\alpha_i \in \mathcal{A}$ for every α_i. In the other direction, if q is satisfiable for AR with a witness $\{\alpha_1\}, ..., \{\alpha_k\}$ such that $\alpha_i \in \mathcal{A}$ for every α_i, $\mathcal{A} \cup \mathcal{P}_+ \cup \bigcup_{i=1}^{k} \{\alpha_i\} = \mathcal{A} \cup \mathcal{P}_+$, so every $q' \in \mathcal{S}_{\mathcal{U}}(\mathcal{P})$ is satisfiable in $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{P}_+ \cup \bigcup_{i=1}^{k} \{\alpha_i\}\rangle$ (using $\mathcal{P}_-$). It follows that q is satisfiable for AR w.r.t. $\mathcal{P}$. $\square$

The following proposition gives the complexity upper bounds of deciding if an answer is satisfied, satisfiable or satisfiable w.r.t. a repair plan for AR.

Proposition 5.4.9. *Deciding if a wanted answer is satisfied or satisfiable for AR is in co*NP. *Deciding if a wanted answer is satisfiable for AR w.r.t. a repair plan is in Σ_2^p, in co*NP *for instance queries.*

Proof. • Showing that $q \in \mathcal{W}$ is *not* satisfied for AR can be done by guessing a repair $\mathcal{R}$ of $\mathcal{K}$ such that every cause for q in $\mathcal{R}$ contains some false assertion, so is in NP.

• Showing $q \in \mathcal{W}$ is *not* satisfiable for AR can be done by guessing a repair $\mathcal{R}$ of the KB $\langle \mathcal{T}, (\mathcal{A} \backslash False_{\mathsf{user}}) \cup True^{rel}_{\mathsf{user}}\rangle$ such that $\langle \mathcal{T}, \mathcal{R}\rangle \not\models q$, so is in NP.

• Deciding whether $q \in \mathcal{W}$ is satisfiable for AR w.r.t. a repair plan $\mathcal{P}$ can be done by guessing a repair plan $\mathcal{P}'$, and checking in Δ_2^p that $\mathcal{P}'$ is validatable, extends $\mathcal{P}$, satisfies all answers satisfied by $\mathcal{P}$ (at most $2 * |\mathcal{W}|$ calls to a coNP oracle), and satisfies q for AR, so is in Σ_2^p.

If q is an instance query, we first check in P if q is satisfiable w.r.t. $\mathcal{P}$ for IAR. If it is, q is also satisfiable for AR w.r.t. $\mathcal{P}$. If it is not, by Lemma 5.4.8, q is satisfiable for AR w.r.t. $\mathcal{P}$ iff it is satisfiable for AR with a witness included in $\mathcal{A}$, which can be check in coNP (show that this is not the case by guessing a repair $\mathcal{R}$ of the KB $\langle \mathcal{T}, (\mathcal{A} \backslash False_{\mathsf{user}})\rangle$ such that $\langle \mathcal{T}, \mathcal{R}\rangle \not\models q$). $\square$

We now establish the complexity of deciding the optimality of a repair plan when AR is taken into account for the satisfaction of wanted answers.

Theorem 5.4.10. *Deciding if a repair plan is globally $\preceq$-optimal is Π_2^p-complete for $\preceq \in \{\preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$ ($\Delta_2^p[O(\log n)]$-complete if all wanted answers are instance queries), and $\Delta_2^p[O(\log n)]$-complete for $\preceq_{\mathcal{W}}$.*

Deciding if a repair plan is locally $\preceq$-optimal is Π_2^p-complete for $\preceq \in \{\preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}\}$ ($\Delta_2^p[O(\log n)]$-complete if all wanted answers are instance queries), and $\Delta_2^p[O(\log n)]$-complete for $\preceq \in \{\preceq_{\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$.

Upper bounds. • For global or local $\preceq_{\mathcal{W}}$-optimality, the conditions of Proposition 5.4.5 can be verified as follows: for each wanted answer that is neither satisfied nor satisfiable for IAR (check in P), use two calls to a coNP oracle that decides if an answer is satisfiable or satisfied for AR. Then check in P if there is one answer such that the oracle has answered yes to the first question and no to the second. Since all calls to the oracle are independent, they can be done in parallel, so the procedure is in $\Delta_2^p[O(\log \text{n})]$ (cf. Appendix A.1).

• For the Π_2^p upper bounds for $\preceq \in \{\preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$, we can show that a repair plan $\mathcal{P}$ is *not* globally (resp. locally) $\preceq$-optimal by guessing a better one (resp. that extends $\mathcal{P}$): guess

$\mathcal{P}'$ (resp. $\mathcal{P}'$ such that $\mathcal{P} \subseteq \mathcal{P}'$), and verify with a polynomial procedure (for the unwanted answers) and $2 * |\mathcal{W}|$ calls to a coNP oracle (for the wanted answers) that $\mathcal{S}(\mathcal{P}) \subset \mathcal{S}(\mathcal{P}')$ (or $\mathcal{S}_{\mathcal{U}}(\mathcal{P}) \subset \mathcal{S}_{\mathcal{U}}(\mathcal{P}')$ in the case $\preceq_{\mathcal{U},\mathcal{W}}$, and $\mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P}) \subset \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P}')$ or $\mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P}) = \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P}')$ and $\mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}) \subset \mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}')$ in the case $\preceq_{\mathcal{W},\mathcal{U}}$).

• When all wanted answers are instance queries, it is possible to decide whether a repair plan $\mathcal{P}$ is globally $\preceq$-optimal for $\preceq \in \{\preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$ in $\Delta^p_2[O(\log \mathrm{n})]$ with the following procedure that use $2 * |\mathcal{W}| + 1$ independent calls to a coNP oracle:

- check whether $\mathcal{P}$ is globally $\preceq$-optimal without taking AR into account (in coNP)

- compute the set $\mathcal{B}$ of wanted answers that are satisfied for AR in $\mathcal{A} \backslash False_{\mathsf{user}}$, i.e. which are satisfiable for AR with a witness included in $\mathcal{A}$ ($|\mathcal{W}|$ calls to a coNP oracle)

- compute $\mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P})$ ($|\mathcal{W}|$ calls to a coNP oracle)

- check that $\mathcal{B} \subseteq \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P})$

We show that $\mathcal{P}$ is globally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal iff it is globally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal without taking AR into account and satisfies for AR every $q \in \mathcal{B}$.

In the first direction, suppose that $\mathcal{P}$ is not globally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal. There exists $\mathcal{P}'$ such that $\mathcal{P} \prec_{\{\mathcal{U},\mathcal{W}\}} \mathcal{P}'$. By Lemma 5.2.5, $\mathcal{P}'' = (False_{\mathsf{user}}, \mathcal{P}'_+)$ is also such that $\mathcal{P} \prec_{\{\mathcal{U},\mathcal{W}\}} \mathcal{P}''$. It follows that $\mathcal{S}_{\mathcal{U}}(\mathcal{P}) \subseteq \mathcal{S}_{\mathcal{U}}(\mathcal{P}'')$, $\mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}) \subseteq \mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}'')$, and $\mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P}) \subseteq \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P}'')$, and one of these inclusions is proper. If $\mathcal{S}_{\mathcal{U}}(\mathcal{P}) \subset \mathcal{S}_{\mathcal{U}}(\mathcal{P}'')$ or $\mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}) \subset \mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}'')$, then $\mathcal{P}$ is not globally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal without taking AR into account. Otherwise, if $\mathcal{S}_{\mathcal{U}}(\mathcal{P}) = \mathcal{S}_{\mathcal{U}}(\mathcal{P}'')$ or $\mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}) = \mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}'')$ and $\mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P}) \subset \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P}'')$, $q \in \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P}'') \backslash \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P})$ is satisfiable for AR with a witness included in $\mathcal{A}$ (otherwise, any witness would contain a true cause and $\mathcal{P}''$ would satisfy q for IAR, so $\mathcal{P}$ would also satisfy q for IAR) so $q \in \mathcal{B}$ and $q \notin \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P})$.

In the other direction:

- If there exists $q \in \mathcal{B}$ and $q \notin \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P})$, $\mathcal{P}' = (False_{\mathsf{user}}, \mathcal{P}_+)$ is such that $\mathcal{S}_{\mathcal{U}}(\mathcal{P}) \subseteq \mathcal{S}_{\mathcal{U}}(\mathcal{P}')$, $\mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}) \subseteq \mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}')$ (by Lemma 5.2.5), and satisfies q for AR. Indeed, q has a witness for AR included in $\mathcal{A}$ so by definition of the witness, $\langle \mathcal{T}, \mathcal{A} \backslash False_{\mathsf{user}} \rangle \models_{\mathrm{AR}} q$, and $\mathcal{P}_+$ does not add conflicts to the assertions of the witness since they are nonfalse.

- If $(\mathcal{P}_-, \mathcal{P}_+)$ is not $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-globally optimal without considering AR because $(\mathcal{P}'_-, \mathcal{P}'_+)$ is a better repair plan, then we show that either $(False_{\mathsf{user}}, \mathcal{P}'_+)$ or $(False_{\mathsf{user}}, \mathcal{P}_+)$ is a better repair plan than $(\mathcal{P}_-, \mathcal{P}_+)$ when AR is taken into account. First, note that since $\mathcal{S}_{\mathcal{U}}(\mathcal{P}) \subseteq \mathcal{S}_{\mathcal{U}}(\mathcal{P}')$, $\mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}) \subseteq \mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}')$, and one of the inclusions is strict, then $\mathcal{S}_{\mathcal{U}}(\mathcal{P}) \subseteq \mathcal{S}_{\mathcal{U}}(False_{\mathsf{user}}, \mathcal{P}'_+)$, $\mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P}) \subseteq \mathcal{S}^{IAR}_{\mathcal{W}}(False_{\mathsf{user}}, \mathcal{P}'_+)$ and one of the inclusions is strict.

If there exists q such that $q \in \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P})$, $q \notin \mathcal{S}^{AR}_{\mathcal{W}}(False_{\mathsf{user}}, \mathcal{P}'_+)$ and $q \in \mathcal{S}^{IAR}_{\mathcal{W}}(False_{\mathsf{user}}, \mathcal{P}_+)$, then $(False_{\mathsf{user}}, \mathcal{P}_+)$ is a better repair plan than $(\mathcal{P}_-, \mathcal{P}_+)$. Indeed, $q \notin \mathcal{S}^{IAR}_{\mathcal{W}}(\mathcal{P})$ otherwise $q \in \mathcal{S}^{IAR}_{\mathcal{W}}(False_{\mathsf{user}}, \mathcal{P}'_+) \subseteq \mathcal{S}^{AR}_{\mathcal{W}}(False_{\mathsf{user}}, \mathcal{P}'_+)$.

Otherwise, if for every q such that $q \in \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P})$, if $q \notin \mathcal{S}^{AR}_{\mathcal{W}}(False_{\mathsf{user}}, \mathcal{P}'_+)$, then $q \notin \mathcal{S}^{IAR}_{\mathcal{W}}(False_{\mathsf{user}}, \mathcal{P}_+)$, then we show that in this case $(False_{\mathsf{user}}, \mathcal{P}'_+)$ is a better repair plan. Indeed, we can show that $\mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P}) \subseteq \mathcal{S}^{AR}_{\mathcal{W}}(False_{\mathsf{user}}, \mathcal{P}'_+)$: suppose for a contradiction that there exists q such that $q \in \mathcal{S}^{AR}_{\mathcal{W}}(\mathcal{P})$ and $q \notin \mathcal{S}^{AR}_{\mathcal{W}}(False_{\mathsf{user}}, \mathcal{P}'_+)$. By assumption $q \notin \mathcal{S}^{IAR}_{\mathcal{W}}(False_{\mathsf{user}}, \mathcal{P}_+)$. It follows that its causes in $\mathcal{A} \cup \mathcal{P}_+$ are in $\mathcal{A}$ (otherwise there would be some cause $\{\alpha\}$ with $\mathsf{user}(\alpha) = \mathsf{true}$ and the answer would be in $\mathcal{S}^{IAR}_{\mathcal{W}}(False_{\mathsf{user}}, \mathcal{P}_+)$). It follows that q is also AR in $\mathcal{A} \cup \mathcal{P}'_+ \backslash False_{\mathsf{user}}$ (a consistent subset that contradicts every cause in $(\mathcal{A}\cup\mathcal{P}'_+)\backslash False_{\mathsf{user}}$ would also contradict every cause in $(\mathcal{A}\cup\mathcal{P}_+)\backslash False_{\mathsf{user}}$): contradiction.

For $\preceq\in\{\preceq_{\mathcal{U},\mathcal{W}},\preceq_{\mathcal{W},\mathcal{U}}\}$, the proof is similar.

• When all wanted answers are instance queries, for $\preceq\in\{\preceq_{\{\mathcal{U},\mathcal{W}\}},\preceq_{\mathcal{U},\mathcal{W}}\}$, deciding if a repair plan $\mathcal{P}$ is locally $\preceq$-optimal can be done in $\Delta_2^p[O(\log n)]$ using the characterization of Proposition 5.4.5 and the complexity results of Proposition 5.4.9 (satisfaction and satisfiability w.r.t. a plan in coNP for instance queries).

• For local $\preceq_{\mathcal{W},\mathcal{U}}$-optimality, the conditions of Proposition 5.4.5 can also be verified with $2*|\mathcal{W}|$ independent coNP calls by Proposition 5.4.9.

□

We next prove the lower bounds.

Proof of $\Delta_2^p[O(\log n)]$-hardness (all cases). The proof is by reduction from the Parity(SAT) problem, where we assume that the formulas are such that φ_{i+1} is unsatisfiable whenever φ_i is unsatisfiable (cf. Appendix A.3). Consider a Parity(SAT) instance given by $\varphi_1,\dots,\varphi_n$. For each i, $1\leq i\leq n$, let $\{c_{i,1},...,c_{i,r(i)}\}$ be the clauses of φ_i over variables $X_i=\{x_{i,1},...,x_{i,h(i)}\}$.

$$\begin{aligned}
\mathcal{T}=&\{U_E\sqsubseteq U, U_O\sqsubseteq U, \exists P^-\sqsubseteq\neg\exists N^-, \exists P\sqsubseteq\neg\exists U^-, \exists N\sqsubseteq\neg\exists U^-, \exists U_E\sqsubseteq\neg A\}\\
\mathcal{A}=&\{P(c_{i,m},x_{i,l})|x_{i,l}\in c_{i,m}\}\cup\{N(c_{i,m},x_{i,l})|\neg x_{i,l}\in c_{i,m}\}\cup\\
&\{U_O(\varphi_i,c_{i,m})\mid i\equiv 1 \bmod 2, 1\leq i\leq n, 1\leq m\leq r(i)\}\cup\\
&\{U_E(\varphi_i,c_{i+1,m})\mid i\equiv 1 \bmod 2, 1\leq i\leq n-1, 1\leq m\leq r(i+1)\}\cup\\
&\{U_E(\varphi_n,a)\mid \text{if } n\equiv 1 \bmod 2\}\cup\\
&\{A(\varphi_i)\mid i\equiv 1 \bmod 2, 1\leq i\leq n\}\\
\mathcal{W}=&\{\exists x U(\varphi_i,x)\mid i\equiv 1 \bmod 2, 1\leq i\leq n\}
\end{aligned}$$

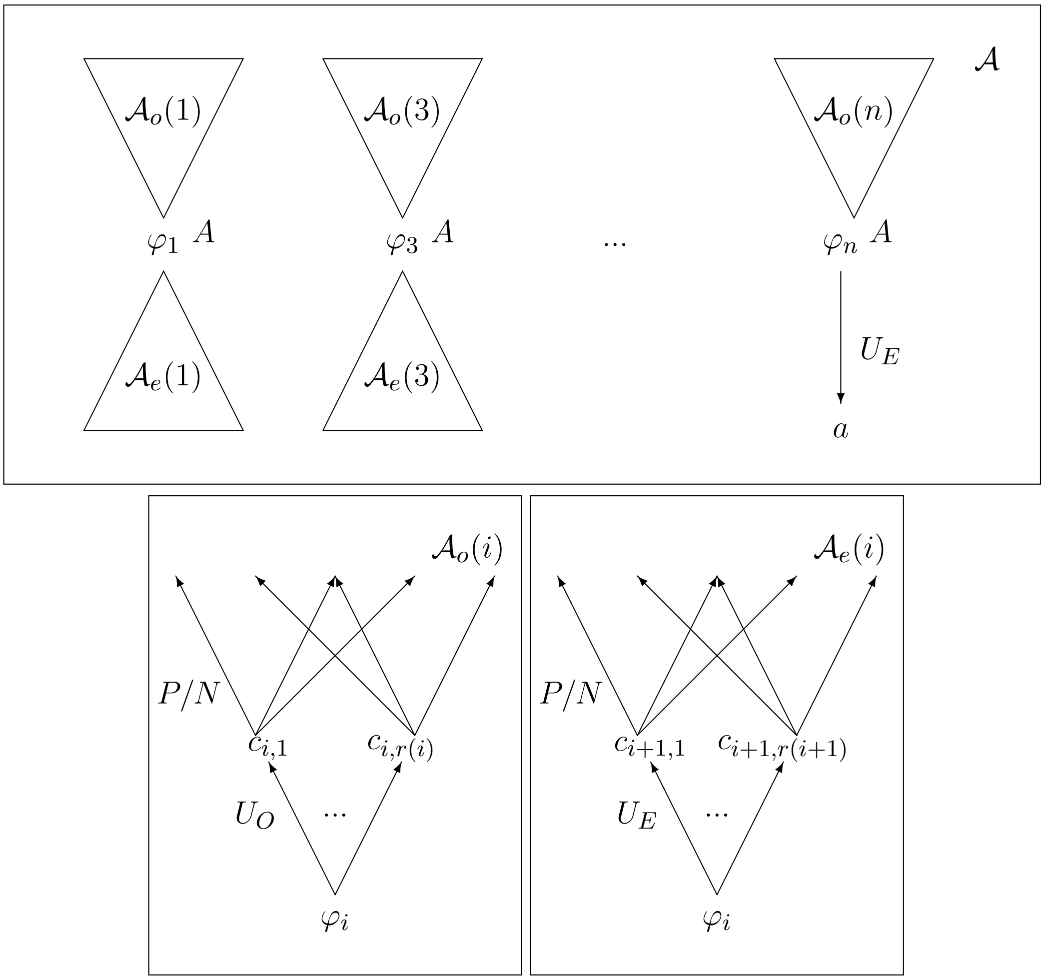
$$\begin{aligned}
True^{rel}_{\mathsf{user}}=&\emptyset\\
False_{\mathsf{user}}=&\{A(\varphi_i)\mid i\equiv 1 \bmod 2, 1\leq i\leq n\}\\
Unk_{\mathsf{user}}=&\mathcal{A}\backslash False_{\mathsf{user}}
\end{aligned}$$

Figure 5.4 illustrates the reduction. We show that $(\mathcal{P}_-,\mathcal{P}_+)=(\emptyset,\emptyset)$ is not optimal iff there exists an odd integer k such that φ_k is satisfiable and φ_{k+1} is unsatisfiable (or n is odd and φ_n is satisfiable).

Note that $(False_{\mathsf{user}},\emptyset)$ is optimal and that the repairs of $\mathcal{A}\backslash False_{\mathsf{user}}$ correspond to the combinations of the possible valuations of the X_i ($1\leq i\leq n$) (that do not interact), and that a formula φ_i with i odd (resp. i even) is satisfiable iff $\exists x U_O(\varphi_i,x)$ (resp. $\exists x U_E(\varphi_{i-1},x)$) is not AR in $\mathcal{A}\backslash False_{\mathsf{user}}$, using the same arguments as usual.

Let $1\leq i\leq n$ be odd. The causes of $\exists x U(\varphi_i,x)$ in $\mathcal{A}$ are the $U_O(\varphi_i,c_{i,m})$ and the $U_E(\varphi_i,c_{i+1,m})$. If φ_i is satisfiable then there exists a repair that contains no $U_O(\varphi_i,c_{i,m})$. There is such a repair that contains $A(\varphi_i)$, so which does not contain any $U_E(\varphi_i,c_{i+1,m})$. Thus $\langle\mathcal{T},\mathcal{A}\rangle\not\models_{\mathsf{AR}}\exists x U(\varphi_i,x)$, and $(\emptyset,\emptyset)$ does not satisfy the wanted answer $\exists x U(\varphi_i,x)$. If φ_{i+1} is unsatisfiable, then $\langle\mathcal{T},\mathcal{A}\backslash False_{\mathsf{user}}\rangle\models_{\mathsf{AR}}\exists x U_E(\varphi_i,x)$, so the wanted answer

Fig. 5.4 Reduction from Parity(SAT) for $\Delta_2^p[O(\log n)]$-hardness of recognition of optimal repair plans using AR semantics. Graphical representation for the case where n is odd.



$\exists x U(\varphi_i, x)$ is satisfiable. It follows that if φ_i is satisfiable and φ_{i+1} is unsatisfiable, then $(False_{\mathsf{user}}, \emptyset)$ is a validatable repair plan which is better than $(\mathcal{P}_-, \mathcal{P}_+)$, so $(\mathcal{P}_-, \mathcal{P}_+)$ is not optimal. If n is odd and φ_n is satisfiable, $(\emptyset, \emptyset)$ does not satisfy $\exists x U(\varphi_n, x)$ but $(False_{\mathsf{user}}, \emptyset)$ satisfies it because $A(\varphi_n)$ is the only conflict of $U_E(\varphi_n, a)$.

In the other direction, if $(\mathcal{P}_-, \mathcal{P}_+)$ is not optimal, there exists $1 \leq i \leq n$ odd such that $(\mathcal{P}_-, \mathcal{P}_+)$ does not satisfy the wanted answer $\exists x U(\varphi_i, x)$, and there exists a validatable repair plan that satisfies it, and in particular $(False_{\mathsf{user}}, \emptyset)$ satisfies it. It follows that φ_i is satisfiable (otherwise the answer $\exists x U(\varphi_i, x)$ is already satisfied) and φ_{i+1} is unsatisfiable (or $i = n$) (otherwise $(False_{\mathsf{user}}, \emptyset)$ would not satisfy it: there would be a repair that corresponds to a valuation of the X_i that satisfies φ_i and a valuation of X_{i+1} that satisfies φ_{i+1} and that would not contains any cause for $\exists x U(\varphi_i, x)$).

□

Proof of Π_2^p-hardness for local or global $\preceq_{\{\mathcal{U},\mathcal{W}\}}$, $\preceq_{\mathcal{U},\mathcal{W}}$ optimality. The proof is by reduction from QBF$_{2,\exists}$. Let $\Phi = \exists x_1 ... x_m \forall y_1 ... y_p \bigvee_{i=1}^{n} C_i$ where $\bigvee_{i=1}^{n} C_i$ is a 2+2 DNF

$(C_i = l_1^i \wedge l_2^i \wedge \neg l_3^i \wedge \neg l_4^i)$.

$$\begin{aligned}
\mathcal{T} =& \{T_Y \sqsubseteq T, F_Y \sqsubseteq F, T_Y \sqsubseteq \neg F_Y\}\\
\mathcal{A} =& \{A(x_j), B(x_j) | 1 \leq j \leq m\} \cup \{T_Y(y_j), F_Y(y_j) \mid 1 \leq j \leq p\} \cup \{A(a), B(a), T(a), F(a)\}\\
&\cup \{P_1(c_i, l_1^i), P_2(c_i, l_2^i), N_1(c_i, l_3^i), N_2(c_i, l_4^i) \mid C_i = l_1^i \wedge l_2^i \wedge \neg l_3^i \wedge \neg l_4^i\}\\
\mathcal{W} =& \{\exists c, l_1, l_2, l_3, l_4\\
&P_1(c, l_1) \wedge P_2(c, l_2) \wedge N_1(c, l_3) \wedge N_2(c, l_4) \wedge T(l_1) \wedge T(l_2) \wedge F(l_3) \wedge F(l_4)\}\\
\mathcal{U} =& \{\exists x T(x) \wedge F(x) \wedge A(x) \wedge B(x)\}
\end{aligned}$$

$$\begin{aligned}
True_{\mathsf{user}}^{rel} =& \{T(x_j), F(x_j) | 1 \leq j \leq m\}\\
False_{\mathsf{user}} =& \{A(a), B(a), T(a), F(a)\}\\
Unk_{\mathsf{user}} =& \mathcal{A} \backslash False_{\mathsf{user}}
\end{aligned}$$

Figure 5.5 illustrates the reduction.

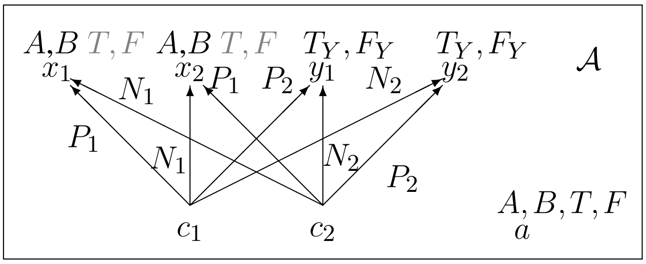
We show that the formula Φ is valid iff the repair plan $(False_{\mathsf{user}}, \emptyset)$ is not optimal. This repair plan satisfies the unwanted answer but does not satisfy the wanted answer. Indeed, the repair of $\mathcal{A} \backslash False_{\mathsf{user}}$ that contains every $F_Y(y_j)$ does not contain any T_Y or T assertion, so does not entail the wanted answer.

If Φ is valid, there exists a valuation ν of the x_j that makes $\forall y_1 ... y_p \bigvee_{i=1}^n \nu(C_i)$ true. Then $(\mathcal{P}_-, \mathcal{P}_+) = (False_{\mathsf{user}}, \{T(x_j) \mid \nu(x_j) = \mathsf{true}\} \cup \{F(x_j) \mid \nu(x_j) = \mathsf{false}\})$ satisfies $\mathcal{U}$ and $\mathcal{W}$. The repairs of $(\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+$ correspond exactly to the valuations of the y_j. Since for every valuation of the y_j, there exists a $C_i = l_1^i \wedge l_2^i \wedge \neg l_3^i \wedge \neg l_4^i$ such that C_i is true, the corresponding repair contains an $F(x_j)$ or $T(x_j)$ per x_j, but not both, as well as $P_1(c_i, l_1^i), P_2(c_i, l_2^i), N_1(c_i, l_3^i), N_2(c_i, l_4^i)$, and $T(l_1^i)$ or $T_Y(l_1^i)$, $T(l_2^i)$ or $T_Y(l_2^i)$, $T(l_3^i)$ or $T_Y(l_3^i)$, and $T(l_4^i)$ or $T_Y(l_4^i)$, depending whether l_j^i is a x_j or a y_j.

In the other direction, if $(False_{\mathsf{user}}, \emptyset)$ is not optimal, then there exists $(\mathcal{P}_-, \mathcal{P}_+)$ that satisfies $\mathcal{U}$ and $\mathcal{W}$. $\mathcal{P}_+$ contains at most one assertion $F(x_j)$ or $T(x_j)$ per x_j. The valuation ν such that $\nu(x_j) = \mathsf{true}$ if $T(x_j) \in \mathcal{P}_+$, $\nu(x_j) = \mathsf{false}$ otherwise is such that $\forall y_1 ... y_p \bigvee_{i=1}^n \nu(C_i)$ is true: since $\mathcal{P}_- \subseteq False_{\mathsf{user}}$, the repairs of $(\mathcal{A} \backslash \mathcal{P}_-) \cup \mathcal{P}_+$ correspond exactly to the valuations of the y_j, so since $\mathcal{W}$ is satisfied for AR, for every valuation ν of the y_j, there is a C_i^ν which evaluates to true in ν.

□

Fig. 5.5 Reduction from QBF$_{2,\exists}$ for Π_2^p-hardness of recognition of $\preceq_{\{\mathcal{U},\mathcal{W}\}}$- or $\preceq_{\mathcal{U},\mathcal{W}}$-optimal repair plans using AR semantics. Graphical representation of the ABox constructed from $\{C_1 = x_1 \wedge y_1 \wedge \neg x_2 \wedge \neg y_2, C_2 = x_2 \wedge y_2 \wedge \neg x_1 \wedge \neg y_1\}$.



Proof for Π_2^p-hardness for global $\preceq_{\mathcal{W},\mathcal{U}}$ optimality. Let $\Phi = \exists x_1...x_m \forall y_1...y_p \bigvee_{i=1}^n C_i$ where $\bigvee_{i=1}^n C_i$ is a 2+2 DNF ($C_i = l_1^i \wedge l_2^i \wedge \neg l_3^i \wedge \neg l_4^i$).

$$
\begin{aligned}
\mathcal{T} =& \{T_Y \sqsubseteq T, F_Y \sqsubseteq F, T_Y \sqsubseteq \neg F_Y, C_1 \sqsubseteq C, C_2 \sqsubseteq C, C_1 \sqsubseteq \neg C_2\}\\
\mathcal{A} =& \{A(x_j), B(x_j) | 1 \leq j \leq m\} \cup \{T_Y(y_j), F_Y(y_j) \mid 1 \leq j \leq p\} \cup \{A(a), B(a), T(a), F(a)\}\\
&\cup \{P_1(c_i, l_1), P_2(c_i, l_2), N_1(c_i, l_3), N_2(c_i, l_4) \mid C_i = l_1^i \wedge l_2^i \wedge \neg l_3^i \wedge \neg l_4^i\} \cup \{C_1(b), C_2(b)\}\\
&\cup \{P_1(e,d), P_2(e,d), N_1(e,d), N_2(e,d)\} \cup \{A(d), B(d)\}\\
\mathcal{W} =& \{\exists c, l_1, l_2, l_3, l_4, x\\
&P_1(c, l_1) \wedge P_2(c, l_2) \wedge N_1(c, l_3) \wedge N_2(c, l_4) \wedge T(l_1) \wedge T(l_2) \wedge F(l_3) \wedge F(l_4) \wedge C(x)\}\\
\mathcal{U} =& \{\exists x T(x) \wedge F(x) \wedge A(x) \wedge B(x)\}
\end{aligned}
$$

$$
\begin{aligned}
True_{\mathsf{user}}^{rel} =& \{T(x_j), F(x_j) | 1 \leq j \leq m\} \cup \{T(d), F(d)\}\\
False_{\mathsf{user}} =& \{A(a), B(a), T(a), F(a)\}\\
Unk_{\mathsf{user}} =& \mathcal{A} \setminus False_{\mathsf{user}}
\end{aligned}
$$

Figure 5.6 illustrates the reduction. We show that Φ is valid iff $(False_{\mathsf{user}}, True_{\mathsf{user}}^{rel})$ is not optimal.

The repair plan $(False_{\mathsf{user}}, True_{\mathsf{user}}^{rel})$ satisfies $q \in \mathcal{W}$ for AR, since the function π such that $\pi(c) = e, \pi(l_1) = \pi(l_2) = \pi(l_3) = \pi(l_4) = d$, and $\pi(x) = b$ is a match for the wanted answer, but $(False_{\mathsf{user}}, True_{\mathsf{user}}^{rel})$ does not satisfy the unwanted answer. Note that the wanted answer cannot be satisfied for IAR because $\exists x C(x)$ has only two causes $C_1(b)$ and $C_2(b)$ which are in conflict and are unknown.

If Φ is valid, it is possible to satisfy both the wanted answer for AR and the unwanted answer as in the previous proof.

If $(False_{\mathsf{user}}, True_{\mathsf{user}}^{rel})$ is not optimal, there exists $(\mathcal{P}_-, \mathcal{P}_+)$ that satisfies both the wanted and unwanted answers (since $(False_{\mathsf{user}}, True_{\mathsf{user}}^{rel}) \prec_{\mathcal{W},\mathcal{U}} (\mathcal{P}_-, \mathcal{P}_+)$ and the wanted answer is not satisfiable for IAR). Then Φ is satisfiable as in previous proof: e and d cannot be used to satisfy the wanted answer, since it would require to add both $T(d)$ and $F(d)$ and would make the unwanted answer not satisfied.

$\square$

Fig. 5.6 Reduction from $QBF_{2,\exists}$ for Π_2^p-hardness of recognition of globally $\preceq_{\mathcal{W},\mathcal{U}}$-optimal repair plans using AR semantics. Graphical representation of the ABox constructed from $\{C_1 = x_1 \wedge y_1 \wedge \neg x_2 \wedge \neg y_2, C_2 = x_2 \wedge y_2 \wedge \neg x_1 \wedge \neg y_1\}$.

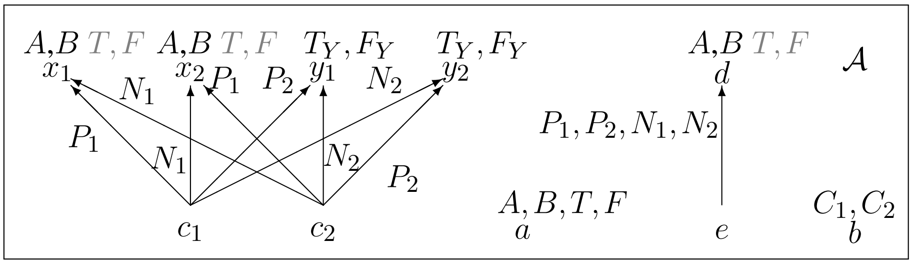


Table 5.1 Recognition of an optimal repair plan in DL-Lite$_\mathcal{R}$: complexity w.r.t. $|\mathcal{A}|$, $|\mathcal{U}|$, $|\mathcal{W}|$, and $|True^{rel}_{\mathsf{user}}|$.

	with AR		**without AR**	
	Global	**Local**	**Global**	**Local**
$\preceq_\mathcal{U}$	in P	in P	in P	in P
$\preceq_\mathcal{W}$	$\Delta_2^p[O(\log n)]$-co*	$\Delta_2^p[O(\log n)]$-co*	in P	in P
$\preceq_{\mathcal{W},\mathcal{U}}$	Π_2^p-co / $\Delta_2^p[O(\log n)]$-co$^{\dagger *}$	$\Delta_2^p[O(\log n)]$-co*	coNP-co*	in P
$\preceq_{\mathcal{U},\mathcal{W}}$	Π_2^p-co / $\Delta_2^p[O(\log n)]$-co$^{\dagger *}$	Π_2^p-co / $\Delta_2^p[O(\log n)]$-co$^{\dagger *}$	coNP-co*	in P
$\preceq_{\{\mathcal{U},\mathcal{W}\}}$	Π_2^p-co / $\Delta_2^p[O(\log n)]$-co$^{\dagger *}$	Π_2^p-co / $\Delta_2^p[O(\log n)]$-co$^{\dagger *}$	coNP-co*	in P

† upper bounds hold for instance queries wanted answers
* lower bounds hold for instance queries wanted answers
If $\mathcal{U}$ contains only instances queries: deciding global or local $\preceq$-optimality for $\preceq \in \{\preceq_{\mathcal{W},\mathcal{U}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\{\mathcal{U},\mathcal{W}\}}\}$ is $\Delta_2^p[O(\log n)]$-co with AR, and in P without AR

Remark 5.4.11. If $\mathcal{U}$ contains only instance queries, deciding if a repair plan is globally $\preceq$-optimal for $\preceq \in \{\preceq_{\{\mathcal{U},\mathcal{W}\}}, \preceq_{\mathcal{U},\mathcal{W}}, \preceq_{\mathcal{W},\mathcal{U}}\}$ is in $\Delta_2^p[O(\log n)]$, since in this case a repair plan $(\mathcal{P}_-, \mathcal{P}_+)$ is globally $\preceq$-optimal iff it is globally $\preceq_\mathcal{U}$-optimal (i.e. $\mathcal{P}_-$ contains all causes of unwanted answers) and globally $\preceq_\mathcal{W}$-optimal. Moreover, the $\Delta_2^p[O(\log n)]$ lower bound holds since the reduction does not use unwanted answers.

Table 5.1 summarizes the complexity results for the recognition of an optimal repair plan.

5.4.2 Discussion: impact of AR semantics on the algorithms

While the complexity of recognizing an optimal repair plan increases when AR is taken into account, the repair plans constructed by the algorithms previously introduced are actually optimal also in this case.

The algorithm $\mathsf{OptimalRepairPlan}_{\mathcal{U}}$ (resp. $\mathsf{OptimalRepairPlan}_{\mathcal{W}}$) still outputs a globally $\preceq_{\mathcal{U}}$ (resp. $\preceq_{\mathcal{W}}$) and locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal repair plan. The only difference when we consider AR is that some answers removed from $\mathcal{W}$, because they cannot be satisfied for IAR semantics, are actually satisfied for AR. Indeed, every known cause that can be added without deteriorating the satisfaction of the unwanted answers has been added, and all of its false conflicts removed. Therefore, the output plan satisfies for IAR (resp. for AR) every wanted answer that is satisfiable for IAR (resp. for AR) w.r.t. the output plan. We could add a step that checks whether an answer not satisfiable for IAR w.r.t. the plan is satisfied for AR (by simply checking if the answer holds under AR semantics, since every false assertion has been removed from its causes), to also provide the information about the satisfied answers for AR.

However, we could also modify the algorithm by dividing the addition step into two phases: the first one tries to satisfy wanted answers for AR, and the second one tries to satisfy for IAR the answers already satisfied for AR. In some cases, this procedure allows us to satisfy more answers for AR than the existing algorithm. Indeed, since we will not add every known cause for an answer already satisfied for AR, we may be able to add causes to satisfy another wanted answer that would have interacted in a wrong way with these superfluous causes. For instance, suppose that $\mathsf{Prof}(a)$ and $\mathsf{GradCourse}(c)$ are wanted, that $\exists xyz \mathsf{Advise}(x,b) \wedge \mathsf{Teach}(x,y) \wedge \mathsf{GradCourse}(y) \wedge \mathsf{TakeCourse}(z,y)$ is unwanted, that AProf and FProf are disjoint, and the ABox contains the assertions $\mathsf{AProf}(a)$, $\mathsf{FProf}(a)$, $\mathsf{Teach}(a,c)$, and $\mathsf{TakesCourse}(c,d)$. Suppose that the user does not know whether these assertions are true or false but knows $\mathsf{Advise}(a,b)$ and $\mathsf{GradCourse}(c)$. If he adds $\mathsf{Advise}(a,b)$ to satisfy $\mathsf{Prof}(a)$ for IAR, he cannot satisfy $\mathsf{GradCourse}(c)$ and the unwanted answer together, whereas if he does not add it, he can satisfy the unwanted answer and every wanted answer, at least for AR. The plans $(\emptyset, \{\mathsf{Advise}(a,b)\})$ and $(\emptyset, \{\mathsf{GradCourse}(c)\})$ are both locally $\preceq_{\{\mathcal{U},\mathcal{W}\}}$-optimal, since extending them cannot improve them, but the second one satisfies more answers for AR.

Regarding deletion-only repair plans, since the potential solutions for IAR are potential solutions for AR and potential solutions for AR can be extended to potential solutions for IAR (cf. Proposition 5.4.1), the sets of relevant, necessarily false or necessarily nonfalse assertions, and the MCSWs are exactly the same as before. As for $\mathsf{OptimalRepairPlan}_{\mathcal{U}}$ and $\mathsf{OptimalRepairPlan}_{\mathcal{W}}$, we can add a step that checks whether the wanted answers that are not satisfied for IAR are satisfied for AR at the end of $\mathsf{OptDeletionRepairPlan}$, to inform the user of the satisfaction of the wanted answers under AR semantics, but the repair plan that is returned is optimal in all cases.

Overall, we believe that the AR semantics can be profitably exploited in the following scenario: a user begins by executing $\mathsf{OptDeletionRepairPlan}$ as a first step to construct a repair plan such that the resulting knowledge base satisfies all satisfiable unwanted answers and satisfies for IAR all wanted answers satisfiable for IAR without inserting new assertions. However, this plan does not satisfy some wanted answers, either because they became non-brave at Step 31, or because the user does not know how to clean a cause (in this case the algorithm ended at Step 17). He therefore uses the insertion step of $\mathsf{OptimalRepairPlan}_{\mathcal{U}}$ (Step 5) to try to satisfy the remaining unsatisfied wanted answers, but considering as satisfied the answers satisfied for AR. Note that in this case the final repair plan is not optimal since it may not satisfy for IAR some wanted answers that are satisfiable for IAR, but the number of insertions as well as the effort of the user is reduced.

5.5 Implementation and experiments

5.5.1 Computing deletion-only repair plans with CQAPri

We implemented the core components of the OptDeletionRepairPlan within CQAPri. During the query answering phase, the causes of the answers are computed and stored as for explaining the answers. The user is then asked to input unwanted and wanted answers. Based on this input and the precomputed causes and conflicts, CQAPri constructs the encoding $\varphi = \varphi_{\mathcal{U}} \wedge \varphi_{\mathcal{W}}$ of Figure 5.2 and uses the SAT solver to check whether there exists a potential solution. If so, it computes the necessarily (non)false assertions by checking for each relevant assertion α if $\varphi \wedge (\neg) x_\alpha$ is unsatisfiable, and sorts the remaining relevant assertions. Their impact is computed as follows: for each other relevant and non necessarily (non)false assertion β, CQAPri checks whether $\varphi \wedge x_\alpha \wedge (\neg) x_\beta$ is unsatisfiable to compute the impact of α being false, and whether $\varphi \wedge \neg x_\alpha \wedge (\neg) x_\beta$ is unsatisfiable for the impact of α being true. For the latter case, the number of conflicts of α in the relevant assertions is also taken into account. To compute the necessarily (non)false assertions and the impact of the others, we rely on a functionality of SAT4J that allows us to assign a value to some variables (for instance assuming that x_α is true and x_β false allows us to decide if removing α implies that β is necessarily false), so that the solver does not begin from scratch at each call but can capitalize on the clauses learned when deciding whether φ is satisfiable and on the model already found.

If the user validates the necessarily (non)false assertions, he may input false and true assertions. The encoding is then updated to take the user input into account by adding clauses of the form x_α if α is false and $\neg x_\alpha$ if it is nonfalse, and CQAPri computes the necessarily (non)false assertions and the impact of the others for the new step.

At each step, CQAPri checks whether the current repair plan is a potential solution by verifying if: (i) it intersects all unwanted causes and (ii) it removes all conflicts of a cause that it does not intersect for each wanted answer. In this case, CQAPri stops and outputs the repair plan. To fully implement OptDeletionRepairPlan, we should add a phase that ensures that each wanted answer has an IAR cause validated by the user, which does not contain any false assertion.

If at some point there does not exist any potential solution (i.e. the encoding is unsatisfiable), CQAPri uses SAT4J to find the MCSWs by computing the maximal satisfiable subsets (MSSes) of the encoding, that are the complements of the MCSes. We did not implement the modification of the wanted answers by the user.

Fig. 5.7 Assertions of the original ABox considered false for building QRPs.

TakeCourse$(Dept2.Univ0/GraduateStudent131, Dept2.Univ0/GraduateCourse8)$
WorkFor$(Dept2.Univ0/AssociateProfessor9, Dept2.Univ0)$
WorkFor$(Dept2.Univ0/AssistantProfessor0, Dept2.Univ0)$
DoctoralDegreeFrom$(Dept2.Univ0/FullProfessor4, University463)$
DoctoralDegreeFrom$(Dept19.Univ56/AssistantProfessor4, University532)$
DoctoralDegreeFrom$(Dept14.Univ52/AssistantProfessor9, University532)$
UnderGraduateDegreeFrom$(Dept7.Univ14/FullProfessor7, University532)$

Table 5.2 Number of false and true answers per query and ABox.

	q5		q7		q8		q10	
	false	true	false	true	false	true	false	true
u100c1	4	6	7	130	3	184	7	286
u100c20	4	6	8	130	4	184	24	286

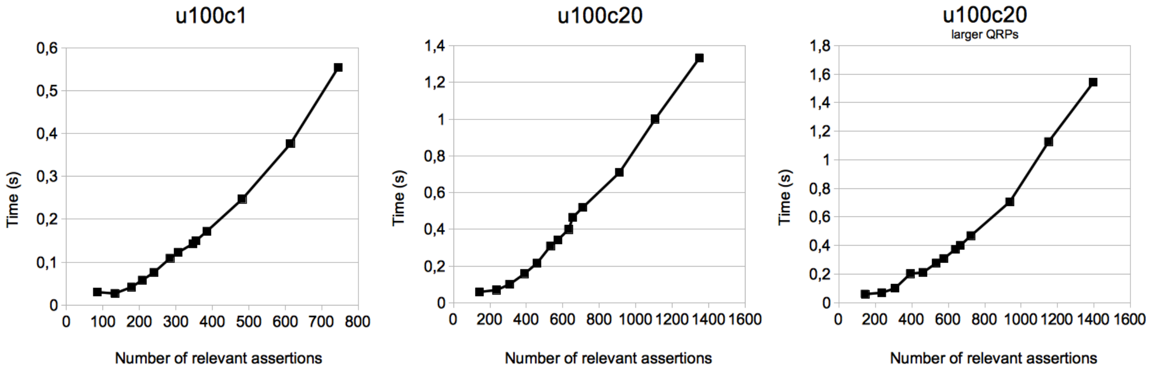
5.5.2 Experimental setting

We focused on measuring the time to decide whether a potential solution exists, to compute the necessarily (non)false assertions, to rank the relevant assertions w.r.t. their impact, and to find the MCSWs.

We built QRPs for ABoxes u100c1 and u100c20 (which have respectively about 5% and 30% of assertions involved in conflicts) using queries q5, q7, q8 and q10 that have many dependent answers (whose causes and conflicts of causes share some assertions).

When building QRPs, the unwanted answers are picked from a set of “false answers” that contains: (i) the answers that were not answers over the initial consistent ABox u100c0, and (ii) the answers such that all their causes contain some assertions that we choose arbitrary and consider to be false. We choose seven such assertions in total (displayed in Figure 5.7). The wanted answers are picked from the complement of these false answers. Table 5.2 shows the number of false and true answers for each query and ABox. We built in sequence 13 QRPs, one being obtained from the preceding QRP by adding further queries answers to $\mathcal{U}$ or $\mathcal{W}$. They have for each of the four queries 1 up to 25 wanted answers and 1 up to 7 unwanted answers. $\mathcal{U} \cup \mathcal{W}$’s size varies from 8 (one wanted and unwanted answer per query) to 102 (the 21 false answers common to u100c1 and u100c20 are unwanted, the 6 true answers of q5 and 25 for each other query are wanted). To study the impact of adding further unwanted answers, we built a second set of QRPs by completing the formers using the additional false answers obtained on u100c20. These QRPs have the same wanted answers and 1 up to 24 unwanted answers. Therefore, $\mathcal{U} \cup \mathcal{W}$’s size is up to 121 (when the 40 false answers on u100c20 are unwanted).

Fig. 5.8 Time in seconds for computing the necessarily false and nonfalse assertions w.r.t. the number of relevant assertions involved in the QRP. The two figures on the left are related to the same QRPs, on u100c1 and u100c20. The QRPs of the figure on the right have more unwanted answers.



5.5.3 Experimental results

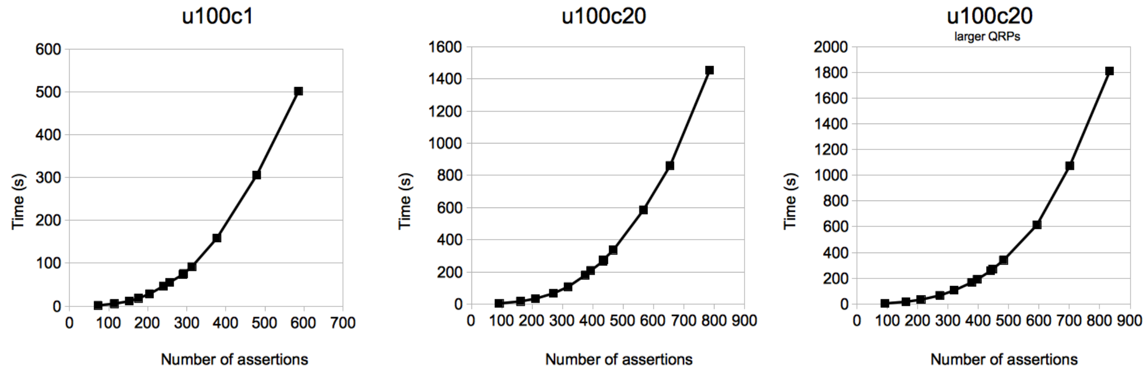
In all of our experiments, deciding if a potential solution exists, as well as computing the relevant assertions, takes a few milliseconds. The difficulty of computing the necessarily (non)false assertions correlates with the number of relevant assertions induced by QRPs. On u100c1, the QRPs involve 85 to 745 relevant assertions, and it takes 30ms to 544ms, while it takes 59ms to 1333ms for the same QRPs on u100c20, where 144 to 1350 assertions are involved, and up to 1541ms for 1395 assertions for the QRPs having more unwanted answers. Figure 5.8 shows the time needed to compute necessarily (non)false assertions w.r.t. the number of relevant assertions on the three cases. While these times seem reasonable in practice, ranking the remaining relevant assertions based on their impact is time consuming (it requires a number of calls to the SAT solver quadratic in the number of assertions): it takes less than 10s up to $\sim$150 assertions, less than 5 minutes up to $\sim$480 assertions, and up to 30 minutes for ranking 833 assertions. Figure 5.9 shows the time needed to rank remaining assertions w.r.t. their number.

The percentage of necessarily (non)false assertions is higher on u100c20 (between 29% and 41% of the relevant assertions) than on u100c1 (between 14% and 21% of the relevant assertions) for the same QRPs as well as for those having more unwanted answers. This is due to higher numbers of necessarily false assertions on u100c20.

Regarding the ranking of the relevant assertions not necessarily (non)false, we observed that the minimal impact guaranteed is typically 0 for our QRPs, but that the sum of the impact of being false or true can be high (up to 20 on u100c20, up to 16 on u100c1). Moreover, the first assertions have an impact significantly higher than the average: on u100c20, for almost every QRP, the 10 first assertions have an impact of 20, while the average ranges from 2.4 to 5.8, and on u100c1, the average impact of the 10 first assertions ranges from 3.8 to 6.5, while the average of all assertions is below 1.

We also executed a few steps of interaction. We accepted the necessarily (non)false assertions, and indicated for the 10 first assertions whether they are true or false. We

Fig. 5.9 Time in seconds for ranking the relevant assertions that are not necessarily false or nonfalse w.r.t. the number of such assertions. The two figures on the left are related to the same QRPs, on u100c1 and u100c20. The QRPs of the figure on the right have more unwanted answers.



observed that during the first steps, the false assertions (the assertions inserted to create conflicts as well as those displayed in Figure 5.7) tend to be ranked in the first tens assertions. Indeed, their impact of being declared true is generally high, either because they have a lot of conflicts, or because they belong to many causes of unwanted answers which have only one remaining assertion not assigned to nonfalse beside them.

We also generate random QRPs to get some problems which have no potential solution. In all cases, computing the MCSWs takes a few milliseconds, and we found at most one MCSW.

6

PREFERRED REPAIR SEMANTICS

In this chapter, we investigate variants of the AR, IAR and brave semantics obtained by replacing the classical notion of repair by one of four different types of preferred repairs. We analyze the complexity of query answering under the resulting semantics, focusing on DL-Lite$_{\mathcal{R}}$. Unsurprisingly, query answering is intractable in all cases, but we nonetheless identify one notion of preferred repair, based upon priority levels, whose data complexity is "only" coNP-complete, as for plain AR semantics. This leads us to propose an approach exploiting a SAT encoding for the semantics based on this kind of repairs. An experimental evaluation of the approach shows that consistent query answering is more difficult with priorities than with the classical set-inclusion repairs, but still scales on realistic cases. The main results of this chapter have been published in [Bienvenu *et al.* 2014].

6.1 Preferred repair semantics

The classical notion of repair integrates a very simple preference relation, namely set inclusion. When additional information on the reliability of ABox assertions is available, it is natural to use this information to identify *preferred repairs*, and to use the latter as the basis of inconsistency-tolerant reasoning. This idea leads us generalize the earlier definitions, using preorders to model preference relations.

Definition 6.1.1. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a KB, and let $\preceq$ be a preorder over subsets of $\mathcal{A}$. A $\preceq$*-repair of* $\mathcal{K}$ is a $\mathcal{T}$-consistent subset of $\mathcal{A}$ which is maximal w.r.t. $\preceq$. The set of $\preceq$-repairs of $\mathcal{K}$ is denoted $Rep_{\preceq}(\mathcal{K})$.

Definition 6.1.2. A Boolean query q is entailed by $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the $\preceq$*-AR semantics*, written $\mathcal{K} \models_{\preceq\text{-AR}} q$, if $\langle \mathcal{T}, \mathcal{R} \rangle \models q$ for every $\mathcal{R} \in Rep_{\preceq}(\mathcal{K})$; it is entailed by $\mathcal{K}$ under the $\preceq$*-IAR semantics*, written $\mathcal{K} \models_{\preceq\text{-IAR}} q$, if $\langle \mathcal{T}, \mathcal{R}_{\cap} \rangle \models q$ where $\mathcal{R}_{\cap} = \bigcap_{\mathcal{R} \in Rep_{\preceq}(\mathcal{K})} \mathcal{R}$; it is entailed by $\mathcal{K}$ under the $\preceq$*-brave semantics*, written $\mathcal{K} \models_{\preceq\text{-brave}} q$, if $\langle \mathcal{T}, \mathcal{R} \rangle \models q$ for some $\mathcal{R} \in Rep_{\preceq}(\mathcal{K})$.

6.1.1 Preference relations

The problem of reasoning on preferred subsets has been studied in a number of areas of AI, such as abduction, belief change, argumentation, and non-monotonic reasoning, see [Eiter & Gottlob 1995, Nebel 1998, Amgoud & Vesic 2011, Brewka *et al.* 2008] and references therein. We consider four standard ways of defining preferences over subsets. Cardinality-maximal repairs are intended for settings in which all ABox assertions are believed to have the same likelihood of being correct. The other three types of preferred repairs target the scenario in which some assertions are considered to be more reliable than others, which can be captured qualitatively by partitioning the ABox into *priority levels* (and then applying either the set inclusion or cardinality criterion to each level), or quantitatively by assigning *weights* to the ABox assertions (and selecting those repairs having the greatest weight).

Cardinality ($\leq$) A first possibility is to compare subsets using set cardinality: $\mathcal{A}_1 \leq \mathcal{A}_2$ iff $|\mathcal{A}_1| \leq |\mathcal{A}_2|$. The resulting notion of $\leq$-repair is appropriate when all assertions are believed to have the same (small) likelihood of being erroneous, in which case repairs with the largest number of assertions are most likely to be correct.

Priority levels ($\subseteq_P$, $\leq_P$) We next consider the case in which ABox assertions have been partitioned into priority levels $P_1, \ldots, P_n$ based on their perceived reliability, with assertions in P_1 considered most reliable, and those in P_n least reliable. Such a prioritization can be used to separate a part of the dataset that has already been validated from more recent additions. Alternatively, one might stratify assertions based upon the concept or role names they use (when some predicates are known to be more reliable), or the data sources from which they originate (in information integration applications). Given a prioritization $P = \langle P_1, \ldots, P_n \rangle$ of $\mathcal{A}$, we can refine the $\subseteq$ and $\leq$ preorders as follows:

- Prioritized set inclusion: $\mathcal{A}_1 \subseteq_P \mathcal{A}_2$ iff $\mathcal{A}_1 \cap P_i = \mathcal{A}_2 \cap P_i$ for every $1 \leq i \leq n$, or there is some $1 \leq i \leq n$ such that $\mathcal{A}_1 \cap P_i \subsetneq \mathcal{A}_2 \cap P_i$ and for all $1 \leq j < i$, $\mathcal{A}_1 \cap P_j = \mathcal{A}_2 \cap P_j$.
- Prioritized cardinality: $\mathcal{A}_1 \leq_P \mathcal{A}_2$ iff $|\mathcal{A}_1 \cap P_i| = |\mathcal{A}_2 \cap P_i|$ for every $1 \leq i \leq n$, or there is some $1 \leq i \leq n$ such that $|\mathcal{A}_1 \cap P_i| < |\mathcal{A}_2 \cap P_i|$ and for all $1 \leq j < i$, $|\mathcal{A}_1 \cap P_j| = |\mathcal{A}_2 \cap P_j|$.

Notice that a single assertion on level P_i is preferred to any number of assertions from P_{i+1}, so these preorders are best suited for cases in which there is a significant difference in the perceived reliability of adjacent priority levels. A few priority classes seems a reasonable scenario.

Weights ($\leq_w$) The reliability of different assertions can be modelled quantitatively using a function $w : \mathcal{A} \to \mathbb{N}$ assigning weights to the ABox assertions. The weight function w induces a preorder $\leq_w$ over subsets of $\mathcal{A}$ in the expected way: $\mathcal{A}_1 \leq_w \mathcal{A}_2$ iff $\sum_{\alpha \in \mathcal{A}_1} w(\alpha) \leq \sum_{\alpha \in \mathcal{A}_2} w(\alpha)$. If the ABox is populated using information extraction techniques, the weights may be derived from the confidence levels output by the extraction tool. Weight-based preorders can also be used in place of the $\leq_P$ preorder to allow for compensation between the priority levels.

Remark 6.1.3. For cardinality, prioritized cardinality or weights, it may be interesting to relax the constraint by considering the repairs of cardinality or weight greater than a fraction of the maximum cardinality or weight to temper the threshold effect. As we will see, this generalization does not cause an increase in complexity.

By definition of $\preceq$-IAR, $\preceq$-AR, and $\preceq$-brave, we still have the following relation:

Proposition 6.1.4. *For every preference relation $\preceq$:*

$$\mathcal{K} \models_{\preceq\text{-}IAR} q \implies \mathcal{K} \models_{\preceq\text{-}AR} q \implies \mathcal{K} \models_{\preceq\text{-}brave} q$$

Moreover, since for $\preceq \in \{\leq, \subseteq_P, \leq_P, \leq_w\}$, every $\preceq$-repair is a $\subseteq$-repair:

Proposition 6.1.5. *For $\preceq \in \{\leq, \subseteq_P, \leq_P, \leq_w\}$, and $S \in \{IAR, AR, brave\}$, if $\mathcal{K} \models_{\subseteq\text{-}S} q$, then $\mathcal{K} \models_{\preceq\text{-}S} q$.*

Finally, since every $\leq_P$-repair is a $\subseteq_P$-repair:

Proposition 6.1.6. *For $S \in \{IAR, AR, brave\}$, if $\mathcal{K} \models_{\subseteq_P\text{-}S} q$, then $\mathcal{K} \models_{\leq_P\text{-}S} q$.*

The following example illustrates the different semantics.

Example 6.1.7. We consider a simple knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$.

$$\begin{aligned}
\mathcal{T} = &\{\mathsf{Student} \sqsubseteq \mathsf{Person}, \mathsf{Prof} \sqsubseteq \mathsf{Person}, \mathsf{Student} \sqsubseteq \neg\mathsf{Prof}, \\
&\exists\mathsf{Teach} \sqsubseteq \mathsf{Prof}, \exists\mathsf{Teach}^- \sqsubseteq \mathsf{Course}, \mathsf{Course} \sqsubseteq \neg\mathsf{Person}\} \\
\mathcal{A} = &\{\mathsf{Student}(ann), \mathsf{Teach}(ann, c_1), \mathsf{Teach}(ann, c_2), \\
&\mathsf{Student}(bob), \mathsf{Prof}(bob), \mathsf{Teach}(c_2, bob), \mathsf{Teach}(bob, c_3)\}
\end{aligned}$$

From $\mathcal{A}$ and the inclusions in $\mathcal{T}$, we can infer that ann is both a student and professor, and bob is a person and course, so the KB is inconsistent as both negative inclusions of $\mathcal{T}$ are violated. The repairs of $\mathcal{K}$ are:

$$\begin{aligned}
\mathcal{R}_1 = &\{\mathsf{Student}(ann), \mathsf{Student}(bob)\} \\
\mathcal{R}_2 = &\{\mathsf{Teach}(ann, c_1), \mathsf{Teach}(ann, c_2), \mathsf{Student}(bob)\} \\
\mathcal{R}_3 = &\{\mathsf{Student}(ann), \mathsf{Prof}(bob), \mathsf{Teach}(bob, c_3)\} \\
\mathcal{R}_4 = &\{\mathsf{Teach}(ann, c_1), \mathsf{Teach}(ann, c_2), \mathsf{Prof}(bob), \mathsf{Teach}(bob, c_3)\} \\
\mathcal{R}_5 = &\{\mathsf{Student}(ann), \mathsf{Teach}(c_2, bob)\} \\
\mathcal{R}_6 = &\{\mathsf{Teach}(ann, c_1), \mathsf{Teach}(ann, c_2), \mathsf{Teach}(c_2, bob)\}
\end{aligned}$$

We observe that the query $\mathsf{Person}(ann)$ is entailed under the AR semantics, since it can be inferred from every repair together with the TBox, but it is not entailed under the IAR semantics, as the intersection of the repairs does not contain any assertion concerning ann.

By moving to preferred repair semantics, we can obtain further answers. First suppose we adopt the cardinality criterion. Then there is a single $\leq$-repair: $\mathcal{R}_4$. Queries $\mathsf{Prof}(ann)$

and $\mathsf{Prof}(bob)$ are entailed under the $\leq$-IAR semantics, while they were not entailed under plain AR semantics.

Next suppose we have the following prioritization $P = \langle P_1, P_2 \rangle$ of $\mathcal{A}$:

$$P_1 = \{\mathsf{Student}(ann), \mathsf{Student}(bob), \mathsf{Prof}(bob)\}$$
$$P_2 = \{\mathsf{Teach}(ann, c_1), \mathsf{Teach}(ann, c_2), \mathsf{Teach}(bob, c_3)\}$$

The $\subseteq_P$-repairs are $\mathcal{R}_1$ and $\mathcal{R}_3$, and there is only one $\leq_P$-repair, namely $\mathcal{R}_3$. Notice that $\mathsf{Student}(ann)$ is entailed under the $\subseteq_P$-IAR and $\leq_P$-IAR semantics, whereas it was not entailed under plain AR semantics, and it conflicts with an assertion entailed under $\leq$-IAR semantics. The assertion $\mathsf{Prof}(bob)$ is entailed under $\leq_P$-IAR semantics, but only $\mathsf{Person}(bob)$ is entailed under $\subseteq_P$-AR semantics.

Finally, if we assign assertions in P_1 a weight of 2, and assertions of P_2 a weight of 1, we obtain two $\leq_w$-repairs: $\mathcal{R}_3$ and $\mathcal{R}_4$. Under $\leq_w$-AR semantics, neither $\mathsf{Prof}(ann)$ nor $\mathsf{Student}(ann)$ is entailed, but only $\mathsf{Person}(ann)$. Under $\leq_w$-IAR semantics, $\mathsf{Prof}(bob)$ is entailed. $\lhd$

6.1.2 Discussion: other notions of prioritized repairs

Three other preference-based semantics are proposed in [Staworko *et al.* 2012] for querying databases that violate integrity constraints, based upon partially ordering the assertions that appear together in a conflict. We show that if such an ordering is induced from an ABox prioritization, then the three semantics all coincide with our $\subseteq_P$-AR semantics. The following definitions are theirs, adapted to our context.

Definition 6.1.8 (Priority)**.** A *priority* $\prec$ is a binary relation on $\mathcal{A}$ such that $\prec$ is acyclic and for every β, δ in $\mathcal{A}$, if $\beta \prec \delta$ then β and δ are in a conflict of $\mathcal{A}$ w.r.t. $\mathcal{T}$.

Definition 6.1.9 (Globally-optimal repair)**.** Given a priority $\prec$, a subset $\mathcal{R}$ of $\mathcal{A}$ is a *globally-optimal repair* if $\mathcal{R} \in Rep_{\subseteq}(\mathcal{T}, \mathcal{A})$ and no nonempty subset $\mathcal{B}$ of assertions from $\mathcal{R}$ can be replaced with a subset $\mathcal{D}$ of $\mathcal{A} \backslash \mathcal{R}$ such that for every $\beta \in \mathcal{B}$ there exists $\delta \in \mathcal{D}$ such that $\beta \prec \delta$ and the resulting set of assertions $(\mathcal{R} \backslash \mathcal{B}) \cup \mathcal{D}$ is $\mathcal{T}$-consistent.

Definition 6.1.10 (Pareto-optimal repair)**.** Given a priority $\prec$, a subset $\mathcal{R}$ of $\mathcal{A}$ is a *Pareto-optimal repair* if $\mathcal{R} \in Rep_{\subseteq}(\mathcal{T}, \mathcal{A})$ and no nonempty subset $\mathcal{B}$ of assertions from $\mathcal{R}$ can be replaced with a subset $\mathcal{D}$ of $\mathcal{A} \backslash \mathcal{R}$ such that for every $\beta \in \mathcal{B}$ and for every $\delta \in \mathcal{D}$, $\beta \prec \delta$ and the resulting set of assertions $(\mathcal{R} \backslash \mathcal{B}) \cup \mathcal{D}$ is $\mathcal{T}$-consistent.

Notice that the definition of Pareto-optimal repair is equivalent if we require $\mathcal{D}$ to be a singleton.

Definition 6.1.11 (Total priority)**.** A priority $\prec$ is *total* if for every β, δ in $\mathcal{A}$, if β and δ are in a conflict together, then $\beta \prec \delta$ or $\delta \prec \beta$.

Definition 6.1.12 (Completion-optimal repair)**.** Given a priority $\prec$, a subset $\mathcal{R}$ of $\mathcal{A}$ is a *completion-optimal repair* if there exists a total priority $\prec'$ such that $\prec \subseteq \prec'$ (i.e. for every β, δ in $\mathcal{A}$, if $\beta \prec \delta$ then $\beta \prec' \delta$) and $\mathcal{R}$ is globally-optimal w.r.t. $\prec'$.

In the following, the sets of repairs of $\mathcal{A}$ for these three notions will be denoted $GRep_{\prec}(\mathcal{T},\mathcal{A})$, $PRep_{\prec}(\mathcal{T},\mathcal{A})$ and $CRep_{\prec}(\mathcal{T},\mathcal{A})$ respectively.

Remark 6.1.13. If the priority $\prec$ is total, then it specifies how to resolve every conflict. In this case, $GRep_{\prec}(\mathcal{T},\mathcal{A}) = PRep_{\prec}(\mathcal{T},\mathcal{A}) = CRep_{\prec}(\mathcal{T},\mathcal{A})$ and contain only one repair [Staworko *et al.* 2012]. This repair is obtained by repeating the following step until the ABox is empty: choose an assertion which is not dominated for $\prec$ by any other, add it to the repair if it does not lead to $\mathcal{T}$-inconsistency, and remove it from $\mathcal{A}$.

These notions use another type of priority than the $\subseteq_P$-preference. It is thus natural to wonder how they are related. We show that the priority induced by a partition of $\mathcal{A}$ into priority levels (and even in the case where some assertions are not assigned to any level) makes the globally-optimal, the Pareto-optimal and the completion-optimal repairs coincide with the $\subseteq_P$-repairs. Notice that in the general case, these three notions of repairs can be different.

Proposition 6.1.14. *Let $P \cup \mathcal{I} = \langle P_1, ..., P_k\rangle \cup \mathcal{I}$ be a partition of $\mathcal{A}$. Let $\prec_P$ be the priority defined by: for β, δ in $\mathcal{A}$, $\beta \prec_P \delta$ if there exist $i, j \in \{1, ..., k\}$ such that $\beta \in P_i$, $\delta \in P_j$ and $i > j$ and β, δ are in a conflict. In this case, $CRep_{\prec_P}(\mathcal{T},\mathcal{A}) = GRep_{\prec_P}(\mathcal{T},\mathcal{A}) = PRep_{\prec_P}(\mathcal{T},\mathcal{A}) = Rep_{\subseteq_P}(\mathcal{T},\mathcal{A})$.*

Proof. [Staworko *et al.* 2012] has shown $CRep_{\prec_P}(\mathcal{T},\mathcal{A}) \subseteq GRep_{\prec_P}(\mathcal{T},\mathcal{A}) \subseteq PRep_{\prec_P}(\mathcal{T},\mathcal{A})$ in the general case.

• $PRep_{\prec_P}(\mathcal{T},\mathcal{A}) \subseteq CRep_{\prec_P}(\mathcal{T},\mathcal{A})$:

Let $\mathcal{R} \in PRep_{\prec_P}(\mathcal{T},\mathcal{A})$. Let $\prec'_P$ be a total extension of the priority $\prec''_P$ defined by: $\beta \prec''_P \delta$ if β, δ are in a conflict and

- there exist $i, j \in \{1, ..., k\}$ such that $\beta \in P_i$, $\delta \in P_j$ and $i > j$, or
- there exists $i \in \{1, ..., k\}$ such that $\beta \in P_i$ and $\delta \in \mathcal{I}$, or
- there exists $i \in \{1, ..., k\}$ such that $\beta \in P_i \backslash \mathcal{R}$ and $\delta \in P_i \cap \mathcal{R}$, or
- $\beta \in \mathcal{I} \backslash \mathcal{R}$ and $\delta \in \mathcal{I} \cap \mathcal{R}$

To get a total priority, it suffices to arbitrary order the assertions belonging to $\mathcal{I} \backslash \mathcal{R}$, to $\mathcal{I} \cap \mathcal{R}$, to $P_i \backslash \mathcal{R}$ and to $P_i \cap \mathcal{R}$ for each i.

Clearly $\prec_P \subseteq \prec'_P$. We show that $\mathcal{R}$ is a globally-optimal repair w.r.t. $\prec'_P$. Suppose it is not. Then there exist nonempty $\mathcal{B} \subseteq \mathcal{R}$ and $\mathcal{D} \subseteq \mathcal{A} \backslash \mathcal{R}$ such that for all $\beta \in \mathcal{B}$, there exists $\delta \in \mathcal{D}$ such that $\beta \prec'_P \delta$ and $(\mathcal{R} \backslash \mathcal{B}) \cup \mathcal{D}$ is $\mathcal{T}$-consistent. If $\mathcal{B} \cap \mathcal{I} \neq \emptyset$, let $\beta_0 \in \mathcal{B} \cap \mathcal{I}$. There exists $\delta_0 \in \mathcal{D}$ such that $\beta_0 \prec'_P \delta_0$. It follows from the construction of $\prec'_P$ that $\delta_0 \notin P_j$ for any j, so $\delta_0 \in \mathcal{I}$. Thus $\delta_0 \in \mathcal{I} \backslash \mathcal{R}$ and $\beta_0 \in \mathcal{I} \cap \mathcal{R}$ so $\delta_0 \prec'_P \beta_0$. This yields a contradiction so $\mathcal{B} \cap \mathcal{I} = \emptyset$. Let $m = min\{i \mid \mathcal{B} \cap P_i \neq \emptyset\}$. If there exist $j < m$ and $\delta_0 \in \mathcal{D} \cap P_j$ then $\beta \prec_P \delta_0$ for all $\beta \in \mathcal{B}$ and since $(\mathcal{R} \backslash \mathcal{B}) \cup \mathcal{D}$ is $\mathcal{T}$-consistent, $(\mathcal{R} \backslash \mathcal{B}) \cup \{\delta_0\}$ is $\mathcal{T}$-consistent. It follows that $\mathcal{R}$ is not a Pareto-optimal repair. Hence for all $\delta \in \mathcal{D}$, there exists $j \geq m$ such that $\delta \in P_j$. Let $\beta_m \in \mathcal{B} \cap P_m$ and $\delta_m \in \mathcal{D}$ be such that $\beta_m \prec'_P \delta_m$. If $\delta_m \in P_j$ and $j > m$, then $\delta_m \prec'_P \beta_m$, else, $\delta_m \in P_m \backslash \mathcal{R}$ and $\delta_m \prec'_P \beta_m$ since $\beta_m \in P_m \cap \mathcal{R}$. In both cases, we obtain a contradiction. Hence $\mathcal{R}$ is a globally-optimal repair w.r.t. $\prec'_P$ and $\mathcal{R} \in CRep_{\prec_P}(\mathcal{T},\mathcal{A})$.

- $Rep_{\subseteq_P}(\mathcal{T},\mathcal{A}) \subseteq PRep_{\prec_P}(\mathcal{T},\mathcal{A})$:

  Let $\mathcal{R} \in Rep_{\subseteq_P}(\mathcal{T},\mathcal{A})$. Suppose for a contradiction that $\mathcal{R} \notin PRep_{\prec_P}(\mathcal{T},\mathcal{A})$. Then we can find nonempty $\mathcal{B} \subseteq \mathcal{R}$ and $\delta \in \mathcal{A}\backslash\mathcal{R}$ such that for all $\beta \in \mathcal{B}$, $\beta \prec_P \delta$ and $(\mathcal{R}\backslash\mathcal{B}) \cup \{\delta\}$ is $\mathcal{T}$-consistent. Since there exists β such that $\beta \prec_P \delta$, then there exists m such that $\delta \in P_m$ by definition of $\prec_P$. For all $\beta \in \mathcal{B}$, since $\beta \prec_P \delta$, there exists i such that $\beta \in P_i$ and $i > m$. Let $\mathcal{A}'$ be any $\mathcal{T}$-consistent extension of $(\mathcal{R}\backslash\mathcal{B}) \cup \{\delta\}$. Since for all $\beta \in \mathcal{B}$, $\beta \in P_i$ with $i > m$, then $\mathcal{R} \cap \mathcal{I} \subseteq \mathcal{A}' \cap \mathcal{I}$, $\mathcal{R} \cap P_j \subseteq \mathcal{A}' \cap P_j$ for all $j < m$ and $\mathcal{R} \cap P_m \subsetneq \mathcal{A}' \cap P_m$, which contradicts the assumption that $\mathcal{R} \in Rep_{\subseteq_P}(\mathcal{T},\mathcal{A})$.

- $PRep_{\prec_P}(\mathcal{T},\mathcal{A}) \subseteq Rep_{\subseteq_P}(\mathcal{T},\mathcal{A})$:

  Let $\mathcal{R} \in PRep_{\prec_P}(\mathcal{T},\mathcal{A})$. Suppose for a contradiction that $\mathcal{R} \notin Rep_{\subseteq_P}(\mathcal{T},\mathcal{A})$. Then there exists a $\mathcal{T}$-consistent subset $\mathcal{A}' \subseteq \mathcal{A}$ and $m \in \{1,...,k\}$ such that $\mathcal{R} \cap \mathcal{I} \subseteq \mathcal{A}' \cap \mathcal{I}$, $\mathcal{R} \cap P_j \subseteq \mathcal{A}' \cap P_j$ for all $j < m$, and $\mathcal{R} \cap P_m \subsetneq \mathcal{A}' \cap P_m$. (Indeed, if $\mathcal{R} \cap P_j \subseteq \mathcal{A}' \cap P_j$ for all j, and $\mathcal{R} \cap \mathcal{I} \subsetneq \mathcal{A}' \cap \mathcal{I}$, then $\mathcal{R} \subsetneq \mathcal{A}'$ is not maximal.) Let $\delta \in (\mathcal{A}' \cap P_m)\backslash\mathcal{R}$ and let C be the set of assertions in $\mathcal{R} \cup \{\delta\}$ which belong to a conflict. The assertion δ belongs to C, since $\mathcal{R}$ is $\mathcal{T}$-consistent and maximal for set inclusion. Let $\mathcal{B} = C\backslash\mathcal{A}'$. $\mathcal{B}$ is nonempty, since $\mathcal{A}'$ is $\mathcal{T}$-consistent and contains δ. We know that if $\beta \in \mathcal{B}$ then there exists $i > m$ such that $\beta \in P_i$ (since for all $\beta \in \mathcal{R}$, if $\beta \in \mathcal{I}$ or $\beta \in P_j$ with $j \leq m$, $\beta \in \mathcal{A}'$). So we have that for all $\beta \in \mathcal{B}$, $\beta \prec_P \delta$ and $(\mathcal{R}\backslash\mathcal{B}) \cup \{\delta\}$ is $\mathcal{T}$-consistent (since all conflicts of $\mathcal{R} \cup \{\delta\}$ are in C, and $C\backslash\mathcal{B} \subseteq \mathcal{A}'$ is $\mathcal{T}$-consistent) which contradicts $\mathcal{R} \in PRep_{\prec_P}(\mathcal{T},\mathcal{A})$.

□

6.2 Complexity analysis

In this section, we study the complexity of query entailment under preferred repair semantics, focusing on DL-Lite$_\mathcal{R}$. However, many of our results hold also for other DLs and ontology languages.

There are some previous works that study the complexity of reasoning with preferred repairs: [Du *et al.* 2013] considers query answering under $\leq_w$-AR semantics for the expressive DL $\mathcal{SHIQ}$, but they focus on ground CQs, as such queries are better-supported by $\mathcal{SHIQ}$ reasoners. By contrast, we work with DL-Lite$_\mathcal{R}$ and can thus use query rewriting techniques to handle CQs with existential variables. In [Lopatenko & Bertossi 2007], the authors study the complexity of query answering in the presence of denial constraints in the databases setting under the $\leq$-AR and $\leq_w$-AR semantics. Because of the difference in setting, we could not transfer their complexity results to DL-Lite$_\mathcal{R}$.

Table 6.1 recalls existing results for query entailment under the standard AR, IAR and brave semantics and presents our new results for the different preferred repair semantics.

Theorem 6.2.1. *The results in Table 6.1 hold.*

We break the proof of Theorem 6.2.1 down into several propositions. We first consider combined complexity and the AR family of semantics.

Proposition 6.2.2. *Regarding combined complexity, CQ entailment over DL-Lite$_\mathcal{R}$ KBs is Π^p_2-complete under $\leq$-AR, $\subseteq_P$-AR, $\leq_P$-AR, and $\leq_w$-AR semantics.*

Table 6.1 Data and combined complexity of CQ entailment over DL-Lite$_{\mathcal{R}}$ KBs under AR, IAR, and brave semantics for different types of preferred repairs. For instance queries, the data and combined complexity coincide with the data complexity for CQs. All results are completeness results unless otherwise noted. † $\Delta_2^p[O(\log \text{n})]$-complete under the assumption that there is a bound on the number of priority classes (resp. maximal weight).

	$\subseteq$	$\leq$	$\subseteq_P$	$\leq_P$	$\leq_w$
AR	coNP	$\Delta_2^p[O(\log \text{n})]$	coNP	Δ_2^p †	Δ_2^p †
IAR	in P	$\Delta_2^p[O(\log \text{n})]$	coNP	Δ_2^p †	Δ_2^p †
brave	in P	$\Delta_2^p[O(\log \text{n})]$	NP	Δ_2^p †	Δ_2^p †

Data complexity

	$\subseteq$	$\leq$	$\subseteq_P$	$\leq_P$	$\leq_w$
AR	Π_2^p	Π_2^p	Π_2^p	Π_2^p	Π_2^p
IAR	NP	$\Delta_2^p[O(\log \text{n})]$	$\Delta_2^p[O(\log \text{n})]$	Δ_2^p †	Δ_2^p †
brave	NP	$\Delta_2^p[O(\log \text{n})]$	NP	Δ_2^p †	Δ_2^p †

Combined complexity

Proof. First, we observe that for all four notions of preferred repair, it is possible to test in coNP whether a given set constitutes a preferred repair. Indeed, if a consistent subset of the ABox is not a repair, then this is witnessed by another consistent subset which is preferred to it. Thus, non-entailment of a CQ q from a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ can be shown by guessing a subset $\mathcal{R} \subseteq \mathcal{A}$ and using an NP oracle to verify that (i) $\mathcal{R}$ is a preferred repair of $\mathcal{K}$, and (ii) $\langle \mathcal{T}, \mathcal{R} \rangle \not\models q$.

For the lower bounds, we note that the proof of Π^p_2-hardness of CQ entailment under plain AR semantics can be reused for the $\leq$-AR semantics, since the $\leq$-repairs and $\subseteq$-repairs coincide for the KBs employed in that reduction. The lower bounds for the other semantics follow immediately. □

We next turn to the data complexity of query entailment under the different brave and AR-based semantics.

Proposition 6.2.3. *Regarding data complexity, instance queries and CQ entailment over DL-Lite$_\mathcal{R}$ KBs are co*NP*-complete for $\subseteq_P$-AR, and* NP*-complete for $\subseteq_P$-brave semantics. For instance queries, we obtain co*NP*-completeness also for combined complexity.*

Proof. We observe that it can be tested in polynomial time (w.r.t. combined complexity) whether a subset $\mathcal{R} \subseteq \mathcal{A}$ is a $\subseteq_P$-repair. This can be done by first verifying that $\mathcal{R}$ is $\mathcal{T}$-consistent and then for $1 \leq i \leq k$, checking that it is not possible to add an assertion belonging to $P_i \setminus \mathcal{R}$ to $\mathcal{R} \cap (P_1 \cup ... \cup P_i)$ while staying $\mathcal{T}$-consistent. It follows that in the procedure from the proof of Proposition 6.2.2, properties (i) and (ii) can be verified in P w.r.t. data complexity, yielding a coNP upper bound for CQ entailment under $\subseteq_P$-AR.

For $\subseteq_P$-brave, replacing (ii) by $\langle \mathcal{T}, \mathcal{R} \rangle \models q$ in the procedure from the proof of Proposition 6.2.2 gives a procedure to show $\subseteq_P$-brave entailment of a CQ q. This gives us the NP upper bound.

Finally, we note that in the case of instance queries, property (ii) can be checked in P w.r.t. combined complexity in both cases, and so we obtain the upper bounds also w.r.t. combined complexity.

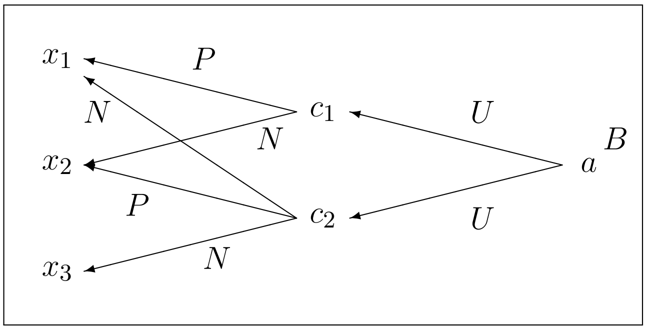
The lower bound for $\subseteq_P$-AR follows from the coNP-hardness of instance query entailment under the standard AR semantics.

For the $\subseteq_P$-brave semantics, the NP-hardness of instance query entailment can be shown by reduction from SAT. Let $\varphi = c_1 \wedge \ldots \wedge c_m$ be a propositional CNF with variables $x_1, ..., x_n$. Consider the TBox and prioritized ABox defined as follows:

$$\begin{aligned}
\mathcal{T} &= \{\exists P^- \sqsubseteq \neg \exists N^-, \exists P \sqsubseteq \neg \exists U^-, \exists N \sqsubseteq \neg \exists U^-, \exists U \sqsubseteq \neg B\} \\
P_1 &= \{P(c_j, x_i) \mid 1 \leq j \leq m, x_i \in c_j\} \cup \{N(c_j, x_i) \mid 1 \leq j \leq m, \neg x_i \in c_j\} \cup \\
&\quad \{U(a, c_j) \mid 1 \leq j \leq m\} \\
P_2 &= \{B(a)\}
\end{aligned}$$

with $\mathcal{A} = P_1 \cup P_2$ and $P = \langle P_1, P_2 \rangle$. Figure 6.1 illustrates the reduction. It can be verified that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\subseteq_P\text{-brave}} B(a)$ iff φ is satisfiable. Indeed, φ is satisfiable iff there exists a

Fig. 6.1 Reduction from SAT for NP-hardness of $\subseteq_P$-brave query answering. Graphical representation of the ABox constructed from an example set of clauses $\varphi = \{C_1 = x_1 \vee \neg x_2, C_2 = \neg x_1 \vee x_2 \vee \neg x_3\}$.



$\subseteq$-repair of P_1 that contains no U assertion (as in the proof for coNP-hardness of plain AR query entailment). Since the U-edges are preferred to $B(a)$, there exists a $\subseteq_P$-repair of $\mathcal{A}$ that contains $B(a)$ iff there exists a $\subseteq$-repair of P_1 that contains no U assertion. It follows that φ is satisfiable iff $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\subseteq_P\text{-brave}} B(a)$.

$\square$

For the $\leq_P$ and $\leq_w$-based semantics, we distinguish two cases, depending on whether the maximal number of priority classes (resp. maximal weight) is considered to be fixed independently of the ABox or is counted as part of the input. The former assumption is made in [Du *et al.* 2013], where a $\Delta_2^p[O(\log \mathrm{n})]$ upper bound is given for the expressive DL $\mathcal{SHIQ}$ that contains DL-Lite$_\mathcal{R}$ as a fragment. A $\Delta_2^p[O(\log \mathrm{n})]$ lower bound for atomic queries is also given in [Du *et al.* 2013], but the proof uses constructs that are unavailable in DL-Lite. Likewise, the proof of $\Delta_2^p[O(\log \mathrm{n})]$-hardness for atomic queries under the $\leq$-AR semantics from [Lopatenko & Bertossi 2007] utilizes denial constraints that cannot be expressed in DL-Lite.

Proposition 6.2.4. *Regarding data complexity, instance query and CQ entailment over DL-Lite$_\mathcal{R}$ KBs are:*

- Δ_2^p*-complete for the* $\leq_P$*-AR,* $\leq_w$*-AR,* $\leq_P$*-brave and* $\leq_w$*-brave semantics,*
- $\Delta_2^p[O(\log n)]$*-complete for the* $\leq$*-AR and* $\leq$*-brave semantics, and for the* $\leq_P$*-AR and* $\leq_P$*-brave, and* $\leq_w$*-AR and* $\leq_w$*-brave semantics, if there is an ABox-independent bound on the number of priority classes (resp. maximal weight).*

For instance queries, these results also hold w.r.t. combined complexity.

For the proof of Proposition 6.2.4 and later propositions, we will leverage the following result, which demonstrates how the semantics based upon the prioritized cardinality preference relation can be recast in terms of weight functions.

Lemma 6.2.5. *(Adapted from [Eiter & Gottlob 1995]) Let* $P = \langle P_1, \ldots, P_k \rangle$ *be a prioritization of* $\mathcal{A}$*, let* $u = (\mathsf{max}_{i=1}^k |P_i|) + 1$*, and let* w *be defined by:* $w(\alpha) = u^{k-i}$ *for* $\alpha \in P_i$*. Then the set of* $\leq_P$*-repairs of* $\langle \mathcal{T}, \mathcal{A} \rangle$ *coincides with the set of* $\leq_w$*-repairs of* $\langle \mathcal{T}, \mathcal{A} \rangle$*, for every TBox* $\mathcal{T}$*.*

The proof of Proposition 6.2.4 will also make frequent use of the following characterization of $\leq$-repairs.

Lemma 6.2.6. *A subset $\mathcal{R}$ of $\mathcal{A}$ is a $\leq$-repair of $\mathcal{A}$ w.r.t. $\mathcal{T}$ if and only if $\mathcal{R}$ is $\mathcal{T}$-consistent and there do not exist subsets X of $\mathcal{R}$ and Y of $\mathcal{A}\backslash\mathcal{R}$ such that $|Y| > |X|$ and $(\mathcal{R}\backslash X) \cup Y$ is $\mathcal{T}$-consistent.*

Proof. For the first direction, let $\mathcal{R}$ be in $Rep_{\leq}(\mathcal{K})$. Suppose for a contradiction that there exist a subset X of $\mathcal{R}$ and a subset Y of $\mathcal{A}\backslash\mathcal{R}$ such that $|Y| > |X|$ and $(\mathcal{R}\backslash X) \cup Y$ is $\mathcal{T}$-consistent. Let $\mathcal{R}' = (\mathcal{R}\backslash X) \cup Y$. Then $\mathcal{R}'$ is a $\mathcal{T}$-consistent subset of $\mathcal{A}$ and $|\mathcal{R}'| = |\mathcal{R}| - |X| + |Y|$, so $|\mathcal{R}'| > |\mathcal{R}|$. Hence, $\mathcal{R}$ is not a $\leq$-repair.

For the other direction, let $\mathcal{R}$ be a $\mathcal{T}$-consistent subset of $\mathcal{A}$ such that there does not exist any subset X of $\mathcal{R}$ such that there exists a subset Y of $\mathcal{A}\backslash\mathcal{R}$ such that $|Y| > |X|$ and $(\mathcal{R}\backslash X) \cup Y$ is $\mathcal{T}$-consistent. Suppose for a contradiction that $\mathcal{R} \notin Rep_{\leq}(\mathcal{K})$. Let $\mathcal{R}' \in Rep_{\leq}(\mathcal{K})$. Since $\mathcal{R}$ is not a $\leq$-repair, $|\mathcal{R}'| > |\mathcal{R}|$. Let $X = \mathcal{R}\backslash\mathcal{R}'$ and $Y = \mathcal{R}'\backslash\mathcal{R}$. Then $(\mathcal{R}\backslash X) \cup Y = \mathcal{R}'$ is $\mathcal{T}$-consistent and $|Y| = |\mathcal{R}'| - |\mathcal{R}' \cap \mathcal{R}|$ and $|X| = |\mathcal{R}| - |\mathcal{R} \cap \mathcal{R}'|$, so $|Y| > |X|$. Hence X, Y contradict the assumption. It follows that $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$. $\square$

We are now ready to give the proof of Proposition 6.2.4.

Proof of Proposition 6.2.4.

Upper bounds. For the $\leq_w$-AR semantics, we use the following procedure to decide whether $\mathcal{K} \not\models_{\leq_w\text{-AR}} q$:

1. Compute the weight u_{rep} of $\leq_w$-repairs by binary search, calling the NP oracle to determine whether there exists a $\mathcal{T}$-consistent subset $\mathcal{A}' \subseteq \mathcal{A}$ such that $\sum_{\alpha\in\mathcal{A}'} w(\alpha) \geq u$ where u is the input.

2. Call the NP oracle to determine whether there exists $\mathcal{R} \subseteq \mathcal{A}$, $\mathcal{T}$-consistent and such that $\sum_{\alpha\in\mathcal{R}} w(\alpha) = u_{\text{rep}}$ and $\langle\mathcal{R},\mathcal{T}\rangle \not\models q$. Return "not entailed" if the call succeeds.

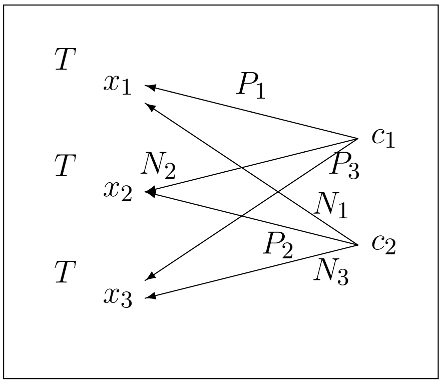
This procedure yields membership in Δ^p_2 for the general case. If there is an ABox-independent bound b on the maximal value of w, then $u_{\text{rep}} \leq \sum_{\alpha\in\mathcal{A}} w(\alpha) \leq b|\mathcal{A}|$, where b is treated as a constant. It follows that the procedure requires only logarithmically many oracle calls, yielding an improved upper bound of $\Delta^p_2[O(\log \text{n})]$. Note that in the case of instance queries, we can test $\langle\mathcal{R},\mathcal{T}\rangle \not\models q$ in polynomial time w.r.t. combined complexity, so these upper bounds apply also w.r.t. combined complexity.

We can derive the upper bounds for the $\leq_P$-AR semantics by applying Lemma 6.2.5. Note that when there is a bound k on the number of priority levels, then this implies a polynomial bound of $(\max_{i=1}^{k}|P_i| + 1)^{k-1} \leq (|\mathcal{A}| + 1)^{k-1}$ on the maximal weight of the corresponding weight function, and so the $\Delta^p_2[O(\log \text{n})]$ upper bound applies. This holds in particular when $k = 1$, i.e. for the $\leq$-AR semantics.

For the $\leq_w$, $\leq_P$, and $\leq$-brave semantics, we simply replace $\langle\mathcal{R},\mathcal{T}\rangle \not\models q$ by $\langle\mathcal{R},\mathcal{T}\rangle \models q$ in the above procedure.

In the case where we consider the repairs that have a weight greater than a fraction $0 < t < 1$ of the maximum weight, we simply need to replace u_{rep} by $t * u_{\text{rep}}$ in the second

Fig. 6.2 Reduction for Δ_2^p-hardness of query answering under $\leq_P$-AR or $\leq_P$-brave semantics. Graphical representation of the ABox constructed from an example set of clauses $\varphi = \{c_1 = x_1 \vee \neg x_2 \vee x_3, c_2 = \neg x_1 \vee x_2 \vee \neg x_3\}$.



step. In this case we cannot use Lemma 6.2.5 to derive the upper bound for $\leq_P$-AR, but it is still possible to use a similar procedure, which starts by computing the number of assertions of each priority in a preferred repair by binary search (compute first the maximum number s^1_{rep} of assertions from P_1 in a consistent subset, then the maximum number of assertions from P_2 in a consistent subset that contains s^1_{rep} of assertions from P_1, etc..).

Δ_2^p lower bound for instance queries under $\leq_P$-AR and $\leq_P$-brave semantics. The proof is by reduction from the following Δ_2^P-complete problem, cf. [Krentel 1988]: given a satisfiable 3CNF formula $\varphi = c_1 \wedge \ldots \wedge c_m$ over variables $x_1, ..., x_n$, decide whether the lexicographically maximum truth assignment satisfying φ with respect to $(x_1, ..., x_n)$, denoted by $\nu_{\max}$, fulfills $\nu_{\max}(x_n) = \mathsf{true}$. We encode this problem as a $\leq_P$-AR (resp. $\leq_P$-brave) query entailment problem as follows:

$$\begin{aligned}
\mathcal{T} =& \{\exists P_\ell \sqsubseteq \neg \exists N_{\ell'}, \exists P_\ell^- \sqsubseteq \neg \exists N_{\ell'}^- \mid 1 \leq \ell, \ell' \leq 3\} \cup \\
& \{\exists P_\ell \sqsubseteq \neg \exists P_{\ell'}, \exists N_\ell \sqsubseteq \neg \exists N_{\ell'} \mid 1 \leq \ell \neq \ell' \leq 3\} \cup \{T \sqsubseteq \neg \exists N_\ell^- \mid 1 \leq \ell \leq 3\} \\
\mathcal{A} =& \{P_\ell(c_j, x_i) \mid x_i \text{ is the } \ell^{th} \text{ literal of } c_j\} \cup \{N_\ell(c_j, x_i) \mid \neg x_i \text{ is the } \ell^{th} \text{ literal of } c_j\} \cup \\
& \{T(x_i) \mid 1 \leq i \leq n\} \\
q =& T(x_n)
\end{aligned}$$

with the prioritization $P = \langle P_1, ..., P_{n+1} \rangle$ of $\mathcal{A}$ as follows: $P_1 = \mathcal{A} \setminus \{T(x_i) \mid 1 \leq i \leq n\}$, and for $1 < i \leq n+1$, $P_i = \{T(x_{i-1})\}$. Figure 6.2 illustrates the reduction.

We show that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\leq_P\text{-AR}} q$ (resp. $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\leq_P\text{-brave}} q$) iff $\nu_{\max}(x_n) = \mathsf{true}$.

The formula φ is satisfiable so there exists a truth assignment ν satisfying φ. Let $\mathcal{A}''$ be the subset of $\mathcal{A}$ defined as follows:

$$\begin{aligned}\mathcal{A}'' = &\{T(x_i) \mid \nu(x_i) = true, 1 \leq i \leq n\} \cup \\ &\Big\{P_l(c_j, x_i) \mid \nu(x_i) = true, 1 \leq j \leq m, x_i\ l^{th} \text{ literal of } c_j\Big\} \cup \\ &\Big\{N_l(c_j, x_i) \mid \nu(x_i) = false, 1 \leq j \leq m, \neg x_i\ l^{th} \text{ literal of } c_j\Big\}\end{aligned}$$

and let $\mathcal{A}'$ be a $\subseteq$-repair of $\mathcal{A}''$.

All the conflicts between the assertions in $\mathcal{A}''$ are of the form $\{P_l(c_j, x_i), P_{l'}(c_j, x_{i'})\}$, $\{N_l(c_j, x_i), N_{l'}(c_j, x_{i'})\}$ or $\{P_l(c_j, x_i), N_{l'}(c_j, x_{i'})\}$ so $\mathcal{A}'$ contains the same assertions involving T than $\mathcal{A}''$ and exactly m assertions of the form $P_l(c_j, x_i)$ or $N_l(c_j, x_i)$ (one for each clause c_j, since there is no reason to remove all $P_l(c_j, x_i)$ or $N_l(c_j, x_i)$ for a given c_j).

It follows that $\mathcal{A}'$ is a $\subseteq$-repair of $\mathcal{A}$ which contains m assertions of P_1. Since it is not possible for a $\mathcal{T}$-consistent subset of $\mathcal{A}$ to contain more than one assertion of P_1 involving c_j for each c_j, a $\leq_P$-repair of $\mathcal{A}$ contains exactly m assertions of P_1.

Then, among all $\subseteq$-repairs of $\mathcal{A}$ which contain m assertions of P_1, if some of them contain $T(x_1)$, the others are not $\leq_P$-repairs of $\mathcal{A}$. In the same way, for all $1 \leq i \leq n$, among all $\subseteq$-repairs of $\mathcal{A}$ which contain m assertions of P_1 and all possible $T(x_h)$ w.r.t. the same condition for $h < i$, if some of them contain $T(x_i)$, the others are not $\leq_P$-repairs of $\mathcal{A}$.

Thus, all $\leq_P$-repairs of $\mathcal{A}$ contain the same assertions $T(x_i)$, and the x_i are precisely the variables assigned to true in the lexicographically maximum truth assignment satisfying φ, ν_m.

It follows that $\mathcal{K} \models_{\leq_P\text{-AR}} q$ (resp. $\mathcal{K} \models_{\leq_P\text{-brave}} q$) if and only if $\nu_m(x_n) = true$.

$\Delta_2^p[O(\log n)]$-hardness for instance queries under $\leq$-AR and $\leq$-brave semantics. The proof is by reduction from the Parity(3SAT) problem where we assume that the formulas are such that φ_{i+1} is unsatisfiable whenever φ_i is unsatisfiable (cf. Appendix A.3). Consider a Parity(3SAT) instance given by $\varphi_1, \ldots, \varphi_n$. For each i, $1 \leq i \leq n$, let $\{c_{i,1}, ..., c_{i,m(i)}\}$ be the clauses of φ_i over variables $X_i = \{x_{i,1}, ..., x_{i,l(i)}\}$. We define an $\leq$-AR (resp. $\leq$-brave) query entailment problem as follows:

$$
\begin{aligned}
\mathcal{T} =& \{\exists A \sqsubseteq Y\} \cup \\
& \{\exists A^- \sqsubseteq \neg V^d, \exists A^- \sqsubseteq \neg W^d \mid 1 \leq d \leq 3\} \cup \\
& \{W^d \sqsubseteq \neg \exists K^m \mid 1 \leq d \leq 3, 1 \leq m \leq 4\} \cup \\
& \{V^d \sqsubseteq \neg \exists E^m \mid 1 \leq d \leq 3, 1 \leq m \leq 4\} \cup \\
& \{\exists K^{m-} \sqsubseteq \neg \exists F^r \mid 1 \leq m \leq 4, 1 \leq r \leq 7\} \cup \\
& \{\exists E^{m-} \sqsubseteq \neg S^g \mid 1 \leq m \leq 4, 1 \leq g \leq 5\} \cup \\
& \{S^g \sqsubseteq \neg \exists F^r \mid 1 \leq g \leq 5, 1 \leq r \leq 7\} \cup \\
& \{\exists F^{r-} \sqsubseteq \neg \exists P^k_l \mid 1 \leq r \leq 7, 1 \leq k \leq 8, 1 \leq l \leq 3\} \cup \\
& \{\exists F^{r-} \sqsubseteq \neg \exists N^k_l \mid 1 \leq r \leq 7, 1 \leq k \leq 8, 1 \leq l \leq 3\} \cup \\
& \{\exists N^{k-}_l \sqsubseteq \neg \exists P^{k'-}_{l'} \mid 1 \leq k, k' \leq 8, 1 \leq l, l' \leq 3\} \cup \\
& \{\exists N^k_l \sqsubseteq \neg \exists P^{k'}_{l'} \mid 1 \leq k, k' \leq 8, 1 \leq l, l' \leq 3\} \cup \\
& \{\exists P^k_l \sqsubseteq \neg \exists P^{k'}_{l'} \mid 1 \leq k, k' \leq 8, 1 \leq l \neq l' \leq 3\} \cup \\
& \{\exists N^k_l \sqsubseteq \neg \exists N^{k'}_{l'} \mid 1 \leq k, k' \leq 8, 1 \leq l \neq l' \leq 3\} \\
\mathcal{A} =& \{A(y, a_i) \mid i \equiv 1 \bmod 2, 1 \leq i \leq n\} \\
& \cup \{K^m(a_i, \varphi_i), W^d(a_i) \mid i \equiv 1 \bmod 2, 1 \leq i \leq n, 1 \leq d \leq 3, 1 \leq m \leq 4\} \\
& \cup \{E^m(a_{i-1}, \varphi_i), V^d(a_{i-1}) \mid i \equiv 0 \bmod 2, 1 \leq i \leq n, 1 \leq d \leq 3, 1 \leq m \leq 4\} \\
& \cup \{S^g(\varphi_i) \mid 1 \leq i \leq n, 1 \leq g \leq 5\} \\
& \cup \{F^r(\varphi_i, c_{i,j}) \mid 1 \leq i \leq n, 1 \leq j \leq m(i), 1 \leq r \leq 7\} \\
& \cup \{P^k_l(c_{i,j}, x_{i,h}) \mid x_{i,h} \text{ is the } l^{th} \text{ literal of } c_{i,j}, 1 \leq i \leq n, 1 \leq l \leq 3, 1 \leq k \leq 8\} \\
& \cup \{N^k_l(c_{i,j}, x_{i,h}) \mid \neg x_{i,h} \text{ is the } l^{th} \text{ literal of } c_{i,j}, 1 \leq i \leq n, 1 \leq l \leq 3, 1 \leq k \leq 8\} \\
q =& Y(y)
\end{aligned}
$$

Note that $\mathcal{T}$, $\mathcal{A}$ and q can be constructed in time polynomial in $\varphi_1, ..., \varphi_n$.

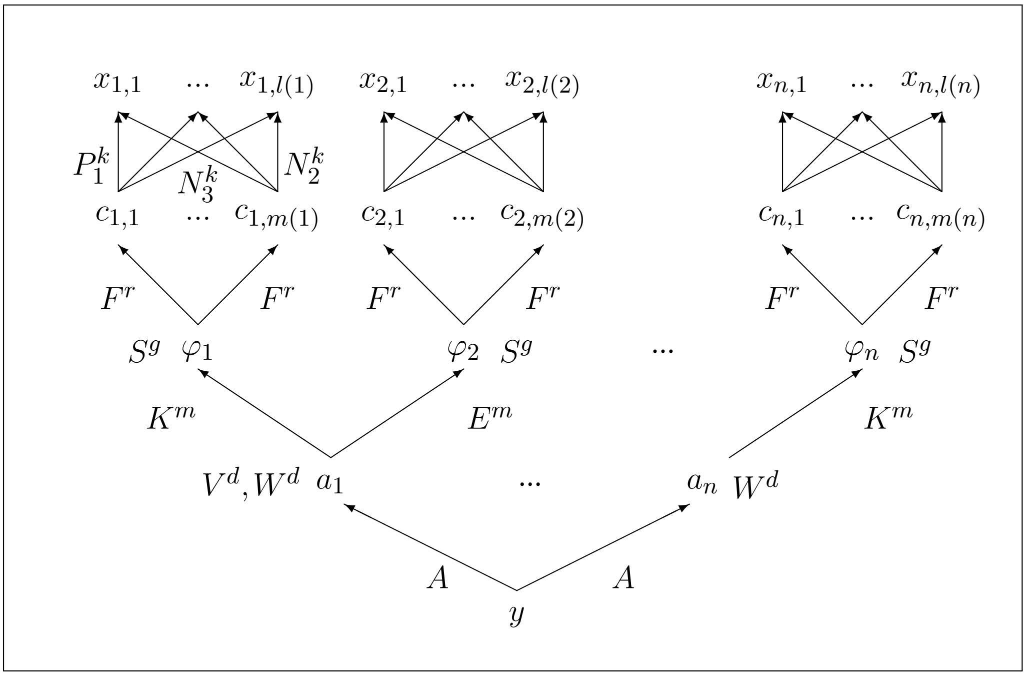
First, notice that if $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$ and $C^p(x) \in \mathcal{R}$ with $C \in \{V, W, S\}$, then $C^{p'}(x) \in \mathcal{R}$ for all $C^{p'}(x) \in \mathcal{A}$. Indeed, since $C^p(x)$ and $C^{p'}(x)$ do not conflict with each other and they conflict with the same assertions of $\mathcal{A}$, both or neither will appear in $\mathcal{R}$. For the same reasons, if $R^p(x,z) \in \mathcal{R}$ with $R \in \{K, E, F, P, N\}$, then $R^{p'}(x,z) \in \mathcal{R}$ for every assertion $R^{p'}(x,z) \in \mathcal{A}$. Hence, if $F^1(\varphi_i, c_{i,j}) \in \mathcal{R}$ for instance, then the seven assertions $F^r(\varphi_i, c_{i,j})$ $(1 \leq r \leq 7)$ are in $\mathcal{R}$, and if $F^1(\varphi_i, c_{i,j}) \notin \mathcal{R}$, then no assertion of the form $F^r(\varphi_i, c_{i,j})$ belongs to $\mathcal{R}$.

Next we establish a series of claims that further characterize the sets in $Rep_{\leq}(\mathcal{K})$.

Claim 1 If φ_i is satisfiable and $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$, then $S^1(\varphi_i) \in \mathcal{R}$.

Proof of claim. Suppose that φ_i is satisfiable and let $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$. Since φ_i is satisfiable, there exists a truth assignment ν_i of X_i such that $\nu_i(\varphi_i) = \text{true}$. It follows that for every

Fig. 6.3 Reduction for $\Delta_2^p[O(\log n)]$-hardness of query answering under $\leq$-AR semantics. Graphical representation of an example ABox (for the case n is odd).



clause $c_{i,j}$ of φ_i, $\nu_i(c_{i,j}) = \mathsf{true}$ so there exists k such that $x_{i,k} \in c_{i,j}$ and $\nu_i(x_{i,k}) = \mathsf{true}$ or $\neg x_{i,k} \in c_{i,j}$ and $\nu_i(x_{i,k}) = \mathsf{false}$. Let

$$\begin{aligned}\mathcal{A}'_{\nu_i} = &\{S^g(\varphi_i) \mid 1 \leq g \leq 5\}\\ &\cup \{P_l^k(c_{i,j}, x_{i,h}) \mid x_{i,h}\ l^{th} \text{ literal of } c_{i,j}, \nu_i(x_{i,h}) = \mathsf{true}, 1 \leq k \leq 8\}\\ &\cup \{N_l^k(c_{i,j}, x_{i,h}) \mid \neg x_{i,h}\ l^{th} \text{ literal of } c_{i,j}, \nu_i(x_{i,h}) = \mathsf{false}, 1 \leq k \leq 8\}\end{aligned}$$

and $\mathcal{A}_{\nu_i}$ be a $\subseteq$-repair of $\mathcal{A}'_{\nu_i}$ w.r.t. $\mathcal{T}$. By construction, the conflicts of $\mathcal{A}'_{\nu_i}$ are of the form $\{P_l^k(c_{i,j}, x_{i,h}), P_{l'}^{k'}(c_{i,j}, x_{i,h})\}$,$\{N_l^k(c_{i,j}, x_{i,h}), N_{l'}^{k'}(c_{i,j}, x_{i,h})\}$ or $\{P_l^k(c_{i,j}, x_{i,h}), N_{l'}^{k'}(c_{i,j}, x_{i,h})\}$. Hence $S^g(\varphi_i) \in \mathcal{A}_{\nu_i}$ $(1 \leq g \leq 5)$ and for each clause $c_{i,j}$, there exists exactly one h such that $P_l^k(c_{i,j}, x_{i,h}) \in \mathcal{A}_{\nu_i}$ or $N_l^k(c_{i,j}, x_{i,h}) \in \mathcal{A}_{\nu_i} (1 \leq k \leq 8)$ (otherwise $\mathcal{A}_{\nu_i}$ would not be maximal for set inclusion).

Suppose for a contradiction that $S^1(\varphi_i) \notin \mathcal{R}$ (thus $S^g(\varphi_i) \notin \mathcal{R}$, $1 \leq g \leq 5$). Let $Y = \mathcal{A}_{\nu_i} \backslash \mathcal{R}$ and let X be the set of the assertions of $\mathcal{R}$ which conflict with some assertion of Y w.r.t. $\mathcal{T}$. By construction of X, $(\mathcal{R} \backslash X) \cup Y$ is $\mathcal{T}$-consistent. Since $S^g(\varphi_i) \in Y, 1 \leq g \leq 5$, and letting n_c be the number of clauses $c_{i,j}$ such that $P_l^k(c_{i,j}, x_{i,h}) \in Y$ or $N_l^k(c_{i,j}, x_{i,h}) \in Y (1 \leq k \leq 8)$, $|Y| = 5 + n_c * 8$. X can contain at most 4 assertions of the form $E^m(a_{i-1}, \varphi_i)$ (if i is even), which conflict with the $S^g(\varphi_i)$, and n_c sets of 7 $F^r(\varphi_i, c_{i,j})$ or of 8 $N_l^k(c_{i,j}, x_{i,h}) \notin Y$ or 8 $P_l^k(c_{i,j}, x_{i,h}) \notin Y$, which conflict with the $P_{l'}^{k'}(c_{i,j}, x_{i,h}) \in Y$ or $N_{l'}^{k'}(c_{i,j}, x_{i,h}) \in Y (1 \leq k' \leq 8)$. Hence $|X| \leq 4 + n_c * 8$. It follows that $|X| < |Y|$, so applying Lemma 6.2.6, we get $\mathcal{R} \notin Rep_{\leq}(\mathcal{K})$. *(End proof of Claim 1)*

Claim 2 If there exists $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$ such that $S^1(\varphi_i) \in \mathcal{R}$, then φ_i is satisfiable.

Proof of claim. Let $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$ be such that $S^1(\varphi_i) \in \mathcal{R}$. For all $c_{i,j}$ and $1 \leq r \leq 7$, $F^r(\varphi_i, c_{i,j}) \notin \mathcal{R}$.

Suppose for a contradiction that there exists $c_{i,j}$ such that for all $x_{i,h}$ and all l, $P_l^1(c_{i,j}, x_{i,h}) \notin \mathcal{R}$, $N_l^1(c_{i,j}, x_{i,h}) \notin \mathcal{R}$. If i is even, it is possible to add 7 assertions of the form $F^r(\varphi_i, c_{i,j})$ and remove the 5 assertions $S^g(\varphi_i) \in \mathcal{R}$ to obtain a $\mathcal{T}$-consistent subset of $\mathcal{A}$, contradicting the fact that $\mathcal{R}$ is a $\leq$-repair. If i is odd, it is possible to add 7 assertions of the form $F^r(\varphi_i, c_{i,j})$ and 3 assertions of the form $W^d(a_i)$ and remove the 5 assertions $S^g(\varphi_i) \in \mathcal{R}$ and 4 assertions $K^m(a_i, \varphi_i) \in \mathcal{R}$ (or add 7 assertions of the form $F^r(\varphi_i, c_{i,j})$ and remove the 5 assertions $S^g(\varphi_i) \in \mathcal{R}$ if there is no assertions of the form $K^m(a_i, \varphi_i)$ in $\mathcal{R}$) to obtain a $\mathcal{T}$-consistent subset of $\mathcal{A}$, contradicting the fact that $\mathcal{R}$ is a $\leq$-repair. Thus for all $c_{i,j}$, there exists $x_{i,h}$ such that $P_l^1(c_{i,j}, x_{i,h}) \in \mathcal{R}$ or $N_l^1(c_{i,j}, x_{i,h}) \in \mathcal{R}$.

Let ν_i be the truth assignment of the variables of X_i defined as follows:

- $\nu_i(x_{i,h}) = \mathsf{true}$ if there exists some assertion $P_l^1(c_{i,j}, x_{i,h}) \in \mathcal{R}$
- $\nu_i(x_{i,h}) = \mathsf{false}$ if there exists some assertion $N_l^1(c_{i,j}, x_{i,h}) \in \mathcal{R}$
- $\nu_i(x_{i,h}) = \mathsf{true}$ otherwise

By construction of ν_i, $\nu_i(c_{i,j}) = \mathsf{true}$ for every clause $c_{i,j}$ of φ_i. It follows that φ_i is satisfiable. *(End proof of Claim 2)*

Claim 3 If φ_i is unsatisfiable and $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$, then there exists j such that $F^1(\varphi_i, c_{i,j}) \in \mathcal{R}$.

Proof of claim. Suppose that φ_i is unsatisfiable and let $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$. Since φ_i is unsatisfiable, $S^g(\varphi_i) \notin \mathcal{R}$, $1 \leq g \leq 5$ (by 2). Hence there exists j such that $F^r(\varphi_i, c_{i,j}) \in \mathcal{R}$ ($1 \leq r \leq 7$) or $E^m(a_{i-1}, \varphi_i) \in \mathcal{R}$ ($1 \leq m \leq 4$) (otherwise $\mathcal{R}$ would not be maximal).

Thus, if i is odd, then there exists j such that $F^r(\varphi_i, c_{i,j}) \in \mathcal{R}$ ($1 \leq r \leq 7$).

If i is even, suppose for a contradiction that $F^r(\varphi_i, c_{i,j}) \notin \mathcal{R}$ ($1 \leq r \leq 7$) for every j. Thus $E^m(a_{i-1}, \varphi_i) \in \mathcal{R}$ ($1 \leq m \leq 4$) so $V^d(a_{i-1}) \notin \mathcal{R}$ ($1 \leq d \leq 3$). Let $X = (\{A(y, a_i)\} \cup \{E^m(a_{i-1}, \varphi_i) \mid 1 \leq m \leq 4\}) \cap \mathcal{R}$ and $Y = \{S^g(\varphi_i) | 1 \leq g \leq 5\} \cup \{V^d(a_{i-1}) \mid 1 \leq d \leq 3\}$. $|X| \leq 5$ and $|Y| = 8$. $X \subseteq \mathcal{R}$, $Y \subseteq \mathcal{A} \backslash \mathcal{R}$, $|Y| > |X|$ and $(\mathcal{R} \backslash X) \cup Y$ is $\mathcal{T}$-consistent so $\mathcal{R}$ is not a $\leq$-repair.

In both cases, there exists j such that $F^r(\varphi_i, c_{i,j}) \in \mathcal{R}$ ($1 \leq r \leq 7$). *(End proof of Claim 3)*

Claim 4 If there exists $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$ such that there exists j such that $F^1(\varphi_i, c_{i,j}) \in \mathcal{R}$, then φ_i is unsatisfiable.

Proof of claim. Let $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$ be such that there exists j such that $F^1(\varphi_i, c_{i,j}) \in \mathcal{R}$. Suppose for a contradiction that φ_i is satisfiable. Then $S^1(\varphi_i) \in \mathcal{R}$ (by 1) which is not possible since $F^1(\varphi_i, c_{i,j})$ and $S^1(\varphi_i)$ are in a conflict. (*End proof of Claim 4*)

Claim 5 Let $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$. If there exists an odd integer k such that $S^1(\varphi_k) \in \mathcal{R}$ and there exists j such that $F^1(\varphi_{k+1}, c_{k+1,j}) \in \mathcal{R}$, or such that $S^1(\varphi_k) \in \mathcal{R}$ and $k = n$, then

$A(y,a_k) \in \mathcal{R}$.

Proof of claim. Let $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$ satisfy the above conditions, and suppose for a contradiction that $A(y,a_k) \notin \mathcal{R}$. Since $\mathcal{R}$ is maximal, $W^d(a_k) \in \mathcal{R}$ $(1 \leq d \leq 3)$ or $V^d(a_k) \in \mathcal{R}$ $(1 \leq d \leq 3)$. Hence, if $k = n$, then $K^m(a_k,\varphi_k) \notin \mathcal{R}$ $(1 \leq m \leq 4)$, and if $k \neq n$, then $K^m(a_k,\varphi_k) \notin \mathcal{R}$ $(1 \leq m \leq 4)$ or $E^m(a_k,\varphi_{k+1}) \notin \mathcal{R}$ $(1 \leq m \leq 4)$.

Let $X = (\{W^d(a_k) \mid 1 \leq d \leq 3\} \cup \{V^d(a_k) | 1 \leq d \leq 3\}) \cap \mathcal{R}$ and $Y = (\{K^m(a_k,\varphi_k) \mid 1 \leq m \leq 4\} \cup \{E^m(a_k) \mid 1 \leq m \leq 4\}) \cap (\mathcal{A} \backslash \mathcal{R})$. If $k = n$ then $|X| \leq 3$ and $|Y| = 4$. If $k \neq n$ then $|Y| \geq 4$ and if $4 \leq |X| \leq 6$ then $|Y| = 8$ (since in this case X contains assertions of the form $W^d(a_k)$ and $V^d(a_k)$). In both cases, we have $X \subseteq \mathcal{R}$, $Y \subseteq \mathcal{A} \backslash \mathcal{R}$, $|Y| > |X|$ and $(\mathcal{R} \backslash X) \cup Y$ is $\mathcal{T}$-consistent, so $\mathcal{R}$ is not a $\leq$-repair.

We have obtained the desired contradiction, so we can conclude that $A(y,a_k) \in \mathcal{R}$. (*End proof of Claim 5*)

Claim 6 Let $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$. If there exists an odd integer k such that $A(y,a_k) \in \mathcal{R}$, then $S^1(\varphi_k) \in \mathcal{R}$ and there exists j such that $F^1(\varphi_{k+1}, c_{k+1,j}) \in \mathcal{R}$, or $S^1(\varphi_k) \in \mathcal{R}$ and $k = n$.

Proof of Claim 6. Let $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$ with $A(y,a_k) \in \mathcal{R}$, for k an odd integer.

First suppose for a contradiction that $S^1(\varphi_k) \notin \mathcal{R}$. Since k is odd, $S^1(\varphi_k)$ cannot be in a conflict with some $E^m(a_k)$, so there exists l such that $F^r(\varphi_k, c_{k,l}) \in \mathcal{R}$ $(1 \leq r \leq 7)$. Hence $K^m(a_k,\varphi_k) \notin \mathcal{R}$ $(1 \leq m \leq 4)$. Let $X = \{A(y,a_k)\}$ and $Y = \{W^d(a_k) \mid 1 \leq d \leq 3\}$. As $X \subseteq \mathcal{R}$, $Y \subseteq \mathcal{A} \backslash \mathcal{R}$, $|Y| > |X|$ and $(\mathcal{R} \backslash X) \cup Y$ is $\mathcal{T}$-consistent, it follows that $\mathcal{R}$ is not a $\leq$-repair. This is a contradiction, so we may conclude that $S^1(\varphi_k) \in \mathcal{R}$.

If $k \neq n$, suppose for a contradiction that for every j, $F^r(\varphi_{k+1}, c_{k+1,j}) \notin \mathcal{R}$ $(1 \leq r \leq 7)$. Since $A(y,a_k) \in \mathcal{R}$, $V^d(a_k) \notin \mathcal{R}$ $(1 \leq d \leq 3)$ so $S^g(\varphi_{k+1}) \in \mathcal{R}$ $(1 \leq g \leq 5)$ and $E^m(a_k,\varphi_{k+1}) \notin \mathcal{R}$ $(1 \leq m \leq 4)$ (otherwise removing the 4 $E^m(a_k,\varphi_{k+1})$ and adding the 5 $S^g(\varphi_{k+1})$ will provide a $\mathcal{T}$-consistent subset of $\mathcal{A}$ with a greater cardinality than $\mathcal{R}$). Let $X = \{A(y,a_k)\}$ and $Y = \{V^d(a_k) \mid 1 \leq d \leq 3\}$. $X \subseteq \mathcal{R}$, $Y \subseteq \mathcal{A} \backslash \mathcal{R}$, $|Y| > |X|$ and $(\mathcal{R} \backslash X) \cup Y$ is $\mathcal{T}$-consistent, so $\mathcal{R}$ is not a $\leq$-repair. Again we have a reached a contradiction, and so may infer that there is some j such that $F^r(\varphi_{k+1}, c_{k+1,j}) \in \mathcal{R}$ $(1 \leq r \leq 7)$. (*End proof of Claim 6*)

We are now ready to show that $\mathcal{K} \models_{\leq\text{-AR}} q$ (resp. $\mathcal{K} \models_{\leq\text{-brave}} q$) if and only if the answer of the initial Parity(SAT) problem is "yes".

- First suppose that there exists an odd integer k such that φ_k is satisfiable and φ_{k+1} is unsatisfiable (or $k = n$). Let $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$. Since φ_k is satisfiable, $S^1(\varphi_k) \in \mathcal{R}$ (by Claim 1), and since φ_{k+1} is unsatisfiable (or $k = n$) there exists j such that $F^1(\varphi_{k+1}, c_{k+1,j}) \in \mathcal{R}$ (by Claim 3) (or $k = n$). Hence $A(y,a_k) \in \mathcal{R}$ (by Claim 5). Thus for every $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$, $A(y,a_k) \in \mathcal{R}$ so $\langle \mathcal{R}, \mathcal{T} \rangle \models Y(y)$. Hence $\mathcal{K} \models_{\leq\text{-AR}} q$ (and therefore $\mathcal{K} \models_{\leq\text{-brave}} q$).
- To show the other direction, suppose that $\mathcal{K} \models_{\leq\text{-brave}} Y(y)$ (or that $\mathcal{K} \models_{\leq\text{-AR}} Y(y)$). There exists $\mathcal{R} \in Rep_{\leq}(\mathcal{K})$ such that $\langle \mathcal{R}, \mathcal{T} \rangle \models Y(y)$, so there exists an (necessarily odd) integer k such that $A(y,a_k) \in \mathcal{R}$. Thus by Claim 6, $S^1(\varphi_k) \in \mathcal{R}$ and there exists j such that $F^1(\varphi_{k+1}, c_{k+1,j}) \in \mathcal{R}$, or $S^1(\varphi_k) \in \mathcal{R}$ and $k = n$. It follows that φ_k is

satisfiable (by Claim 2) and φ_{k+1} is unsatisfiable (by Claim 4) or φ_k is satisfiable and $k = n$.

Remaining lower bounds. Applying Lemma 6.2.5, we can transfer the Δ_2^p lower bound for instance queries under $\leq_P$-AR and $\leq_P$-brave semantics to the $\leq_w$-AR and $\leq_w$-brave semantics. The preceding $\Delta_2^p[O(\log \mathrm{n})]$ lower bound for instance queries under the $\leq$-AR and $\leq$-brave semantics transfers to the $\leq_P$-AR and $\leq_P$-brave semantics (under the bounded priority level assumption). The latter result can then be transferred using Lemma 6.2.5 to the $\leq_w$-AR and $\leq_w$-brave semantics (under the bounded weight assumption). ☐

We next give the combined complexity of query entailment under the different brave semantics.

Proposition 6.2.7. *Regarding combined complexity, CQ entailment over DL-Lite$_\mathcal{R}$ KBs is $\Delta_2^p[O(\log n)]$-complete under $\leq$-brave,* NP*-complete under $\subseteq_P$-brave, and Δ_2^p-complete under $\leq_P$-brave and $\leq_w$-brave semantics. If there is an ABox-independent bound on the number of priority classes (resp. maximal weight), CQ entailment under $\leq_P$-brave (resp. $\leq_w$-brave) is $\Delta_2^p[O(\log n)]$-complete.*

Proof. For the upper bounds, we use the fact that in the decision procedures used in the proofs for data complexity, instead of guessing a repair and checking that it is a preferred repair that entails the query q, we can guess a repair together with a certificate that it entails q, and verify everything in P w.r.t. combined complexity. Therefore the "guess and check" step of all procedures is in NP, and the uppers bounds are the same w.r.t. combined complexity as w.r.t. data complexity. The lower bounds follow from the data complexity. ☐

Finally, we establish the complexity of query entailment under the different IAR semantics.

Proposition 6.2.8. *Regarding data complexity, instance query and CQ entailment over DL-Lite$_\mathcal{R}$ KBs is co*NP*-complete for the $\subseteq_P$-IAR semantics. For instance queries, we also have co*NP*-complete regarding combined complexity.*

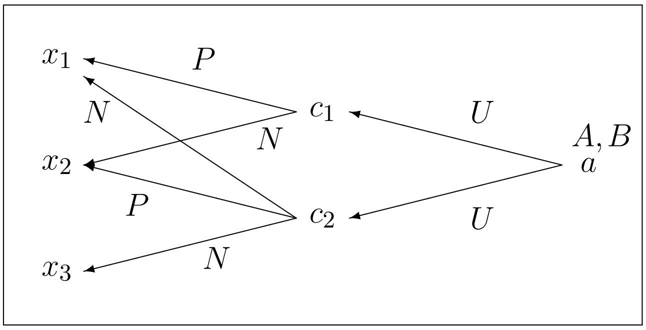
Proof. We can show that $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{\subseteq_P\text{-IAR}} q$ as follows:

1. Guess a subset $\mathcal{A}' = \{\alpha_1, \ldots, \alpha_m\} \subseteq \mathcal{A}$ together with a subset $\mathcal{R}_i \subseteq \mathcal{A}$ with $\alpha_i \notin \mathcal{R}_i$ for each $1 \leq i \leq m$.

2. Verify that (i) each $\mathcal{R}_i$ is a $\subseteq_P$-repair and (ii) $\langle \mathcal{T}, \mathcal{A} \setminus \mathcal{A}' \rangle \not\models q$.

Since $\subseteq_P$-repairs can be identified in polynomial time (cf. proof of Proposition 6.2.3) and query entailment is in P for data complexity, the above procedure runs in non-deterministic polynomial time in the size of $\mathcal{A}$. We thus obtain a coNP upper bound for data complexity. For instance queries, query entailment is in P for combined complexity, so we obtain an coNP upper bound also for combined complexity.

Fig. 6.4 Reduction from UNSAT for coNP-hardness of $\subseteq_P$-IAR query answering. Graphical representation of the ABox constructed from an example set of clauses $\varphi = \{C_1 = x_1 \vee \neg x_2, C_2 = \neg x_1 \vee x_2 \vee \neg x_3\}$.



We show coNP-hardness using a reduction from UNSAT. Let $\varphi = c_1 \wedge \ldots \wedge c_m$ be a propositional CNF with variables $x_1, ..., x_n$. Consider the TBox and prioritized ABox defined as follows:

$$\begin{aligned}
\mathcal{T} &= \{\exists P^- \sqsubseteq \neg \exists N^-, \exists P \sqsubseteq \neg \exists U^-, \exists N \sqsubseteq \neg \exists U^-, \exists U \sqsubseteq \neg B, B \sqsubseteq \neg A\} \\
P_1 &= \{P(c_j, x_i) \mid 1 \leq j \leq m, x_i \in c_j\} \cup \{N(c_j, x_i) \mid 1 \leq j \leq m, \neg x_i \in c_j\} \\
P_2 &= \{U(a, c_j) \mid 1 \leq j \leq m\} \\
P_3 &= \{A(a), B(a)\}
\end{aligned}$$

with $\mathcal{A} = P_1 \cup P_2 \cup P_3$ and $P = \langle P_1, P_2, P_3 \rangle$. Figure 6.4 illustrates the reduction. It can be verified that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\subseteq_P\text{-IAR}} A(a)$ iff φ is unsatisfiable. Indeed, a $\subseteq_P$-repair does not contain $A(a)$ iff it contains $B(a)$ (since it is its only conflict), and since the U-edges are preferred to $B(a)$, a $\subseteq_P$-repair contains $B(a)$ only if it contains some P or N assertions that conflict every U, i.e. if φ is satisfiable. In the other direction, if φ is satisfiable, there exists a $\subseteq_P$-repair that contains no U and such a repair contains either $A(a)$ or $B(a)$. □

Proposition 6.2.9. *For data and combined complexity, instance query and CQ entailment over DL-Lite$_\mathcal{R}$ KBs is:*

- *Δ_2^p-complete for the $\leq_P$-IAR and $\leq_w$-IAR semantics,*
- *$\Delta_2^p[O(\log n)]$-complete for the $\leq$-IAR semantics, and for the $\leq_P$-IAR and $\leq_w$-IAR semantics, if there is an ABox-independent bound on the number of priority classes (resp. maximal weight).*

We also have $\Delta_2^p[O(\log n)]$-completeness for CQ entailment under $\subseteq_P$-IAR semantics for combined complexity.

Proof.

Upper bounds. For the $\leq_w$-IAR semantics, we can use the following procedure to decide whether $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{\leq_w\text{-IAR}} q$:

1. Compute the weight u_{rep} of $\leq_w$-repairs (cf. proof of Proposition 6.2.4).

2. For every $\alpha \in \mathcal{A}$, use an NP oracle to decide if there exists a $\mathcal{T}$-consistent subset $\mathcal{B} \subseteq \mathcal{A}$ such that $\alpha \notin \mathcal{B}$ and $\sum_{\alpha \in \mathcal{B}} w(\alpha) = u_{\text{rep}}$. Let $\mathcal{A}'$ be the set of all assertions for which no such subset exists.

3. Use an NP oracle to verify that the CQ q is not entailed from $\langle \mathcal{T}, \mathcal{A}' \rangle$.

Correctness of the above procedure is straightforward: the ABox $\mathcal{A}'$ constructed in Step 2 is precisely the intersection of the $\leq_w$-repairs. The procedure runs in polynomial time with access to an NP oracle, yielding membership in Δ_2^p. For the bounded weight case, we recall that the class $\Delta_2^p[O(\log n)]$ can be equivalently characterized as the class of decision problems which can be solved in polynomial-time with a single round of parallel calls to an NP oracle, cf. [Buss & Hay 1991]. By using this technique, instead of computing u_{rep} in Step 1 by making a sequence of logarithmically many oracle calls, we can instead issue a single round of parallel calls to the NP oracle. Steps 2 and 3 can be implemented using two further rounds of parallel NP oracle calls. It follows from results in [Buss & Hay 1991] that we can reduce these three rounds into a single one, from which membership in $\Delta_2^p[O(\log n)]$ follows.

Similarly to the proof of Proposition 6.2.4, we can exploit Lemma 6.2.5 to obtain the upper bounds for the $\leq_P$-IAR and $\leq$-IAR semantics. For the $\subseteq_P$-IAR semantics, we can skip Step 1 of the procedure and modify Step 2 by using the polynomial-time procedure for identifying $\subseteq_P$-repairs from the proof of Proposition 6.2.4.

$\Delta_2^p[O(\log n)]$-hardness for CQs and $\subseteq_P$-IAR semantics. The proof is by reduction from the Parity(3SAT) problem. Let $\Phi = \varphi_1, \ldots, \varphi_n$ be a Parity(3SAT) instance satisfying the same restrictions as in the proof of Proposition 6.2.4, where each φ_i is a 3CNF over the variables $x_{i,1}, \ldots, x_{i,g_i}$ composed of m_i clauses $c_{i,1}, \ldots, c_{i,m_i}$. We combine ideas from the proof of Proposition 6.2.8 with a construction from [Bienvenu & Rosati 2013]. In the latter paper, the authors define ABoxes $\mathcal{A}_j$ for the odd $j \in [1,n]$ and a Boolean CQ q with the following properties:

1. if $j_1 \neq j_2$, then $\mathsf{Inds}(\mathcal{A}_{j_1}) \cap \mathsf{Inds}(\mathcal{A}_{j_2}) = \emptyset$;

2. $\langle \emptyset, \bigcup_j \mathcal{A}_j \rangle \models q$ if and only if $\langle \emptyset, \mathcal{A}_j \rangle \models q$ for some j;

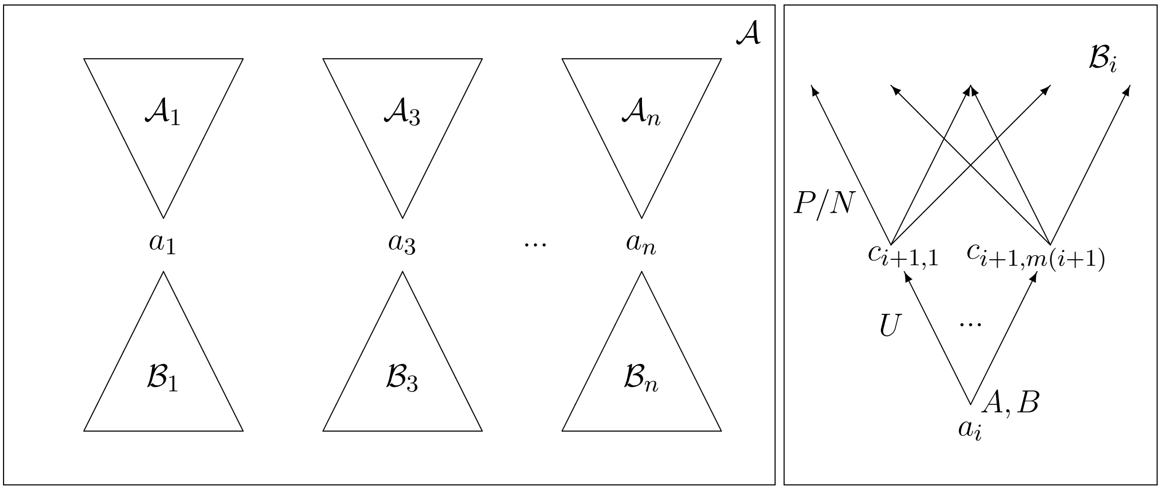
3. $\langle \emptyset, \mathcal{A}_j \rangle \models q$ if and only if φ_ℓ is satisfiable for $1 \leq \ell \leq j$;

4. there is a variable x in q and an individual a_j in each $\mathcal{A}_j$ such that $\langle \emptyset, \mathcal{A}_j \rangle \models q$ if and only if $\langle \emptyset, \mathcal{A}_j \rangle \models q[x \mapsto a_j]$.

The query q and ABoxes $\mathcal{A}_j$ provide a means of showing that the first j formulas are satisfiable, but we still also need a way of showing that the $j+1$st formula is unsatisfiable. To this end, we consider the TBox from the proof of Proposition 6.2.8:

$$\mathcal{T} = \{\exists P^- \sqsubseteq \neg \exists N^-, \exists P \sqsubseteq \neg \exists U^-, \exists N \sqsubseteq \neg \exists U^-, \exists U \sqsubseteq \neg B, B \sqsubseteq \neg A\}$$

Fig. 6.5 Reduction from Parity(SAT) for $\Delta_2^p[O(\log n)]$-hardness of $\subseteq_P$-IAR query answering w.r.t. combined complexity. Graphical representation (case n odd).



where we assume that P,N,U,B,A are fresh roles and concepts, appearing neither in q nor any $\mathcal{A}_j$. We also create ABoxes $\mathcal{B}_j = \mathcal{B}_j^1 \cup \mathcal{B}_j^2 \cup \mathcal{B}_j^3$, for each odd $j \in [1,n]$:

$$\begin{aligned}
\mathcal{B}_j^1 &= \{P(c_{j+1,\ell}, x_{j+1,h}) \mid x_{j+1,h} \in c_{j+1,\ell}\} \cup \{N(c_{j+1,\ell}, x_{j+1,h}) \mid \neg x_{j+1,h} \in c_{j+1,\ell}\} \\
\mathcal{B}_j^2 &= \{U(a_j, c_{j+1,\ell}) \mid 1 \leq \ell \leq m_{j+1}\} \\
\mathcal{B}_j^3 &= \{A(a_j), B(a_j)\}
\end{aligned}$$

where the individuals of the forms $c_{j+1,\ell}$ and $x_{j+1,h}$ are fresh and do not appear in any $\mathcal{A}_j$ or any $\mathcal{B}_{j'}$ for $j' \neq j$, and the individual a_j is shared by $\mathcal{B}_j$ and $\mathcal{A}_j$. We then define the ABox $\mathcal{A}$ as the union of the $\mathcal{A}_j$ and $\mathcal{B}_j$, for all odd $j \in [1,n]$, and consider the following prioritization $P = \langle P_1, P_2, P_3 \rangle$:

$$\begin{aligned}
P_1 &= \bigcup_{\mathsf{odd}\, j \in [1,n]} \mathcal{A}_j \cup \mathcal{B}_j^1 \\
P_2 &= \bigcup_{\mathsf{odd}\, j \in [1,n]} \mathcal{B}_j^2 \\
P_3 &= \bigcup_{\mathsf{odd}\, j \in [1,n]} \mathcal{B}_j^3
\end{aligned}$$

We remark that since the signatures of the ABoxes $\mathcal{A}_j$ and $\mathcal{T}$ are disjoint, the assertions in the ABoxes $\mathcal{A}_j$ are not involved in any conflicts, and hence belong to every $\subseteq_P$ repair. Moreover, because the ABoxes $\mathcal{B}_j$ use disjoint sets of variables, the $\subseteq_P$-repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$ can be obtained by repairing each of the $\mathcal{B}_j$ separately (according to the prioritization P) and taking the union of these repairs and the ABoxes $\mathcal{A}_j$. Using similar arguments to Proposition 6.2.8, one can show that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\subseteq_P\text{-IAR}} A(a_j)$ just in the case that φ_{j+1} is unsatisfiable.

Now let q' be the query obtained by adding $A(x)$ to q. We aim to show that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\subseteq_P\text{-IAR}} q'$ if and only if there is an odd $j \in [1,n]$ such that φ_j is satisfiable and φ_{j+1} is unsatisfiable.

First suppose that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\subseteq_P\text{-IAR}} q'$, and let $\mathcal{R}$ be the intersection of the $\subseteq_P$-repairs. Then $\langle \mathcal{T}, \mathcal{R} \rangle \models q'$, and since $\mathcal{T}$ contains only negative inclusions, we in fact have $\langle \emptyset, \mathcal{R} \rangle \models q'$. Thus, there exists a function π mapping variables in q' to individuals in $\mathcal{R}$ which witnesses the satisfaction of q'. It follows that $A(\pi(x)) \in \mathcal{R}$, and hence $\pi(x) = a_\ell$ for some odd $\ell \in [1,n]$. By above, we know that this means that $\varphi_{\ell+1}$ is unsatisfiable. We have also seen above that $\mathcal{R}$ contains all of the $\mathcal{A}_j$, and since q uses only predicates from the $\mathcal{A}_j$, we must have $\langle \emptyset, \bigcup_j \mathcal{A}_j \rangle \models q$. By Properties 2 and 4, we can infer that $\langle \emptyset, \mathcal{A}_\ell \rangle \models q$. Applying Property 3, we obtain that φ_ℓ is satisfiable.

For the other direction, suppose that φ_j is satisfiable and φ_{j+1} is unsatisfiable. Then by our earlier assumption, for every $1 \leq \ell \leq j$, the formula φ_ℓ is satisfiable. It follows then by Property 3 that $\langle \emptyset, \mathcal{A}_j \rangle \models q$, and so by Property 4, we must have $\langle \emptyset, \mathcal{A}_j \rangle \models q[x \mapsto a_j]$. Since $\mathcal{A}_j$ appears in all repairs, this yields $\langle \mathcal{T}, \mathcal{A}_j \rangle \models_{\subseteq_P\text{-IAR}} q[x \mapsto a_j]$. By earlier arguments, the unsatisfiability of φ_{j+1} means that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\subseteq_P\text{-IAR}} A(a_j)$. Putting this together, we obtain $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\subseteq_P\text{-IAR}} q'$.

Remaining lower bounds. To show the Δ_2^p lower bounds, we can use the reduction from the proof of Proposition 6.2.4: as the TBox does not give any way of deriving $T(x_n)$, the query $T(x_n)$ holds in all $\leq_P$-repairs iff it holds in the intersection of all $\leq_P$-repairs. Finally, a $\Delta_2^p[O(\log \mathrm{n})]$ lower bound for instance query entailment under $\leq$-IAR semantics can be proved similarly to the corresponding result for the $\leq$-AR semantics (Proposition 6.2.4). □

Let us briefly comment on the obtained results. Concerning data complexity, we observe that the use of preferred repairs significantly impacts complexity: we move from polynomial data complexity in the case of (plain) IAR or brave semantics to coNP-hard or NP-hard data complexity (or worse) for IAR and brave semantics based on preferred repairs. Actually, the IAR semantics is just as difficult as the AR semantics. This is due to the fact that there is no simple way of computing the intersection of preferred repairs, whereas this task is straightforward for $\subseteq$-repairs. In the same way, for brave semantics, verifying the existence of a cause for the query is not sufficient to show that it holds under $\preceq$-brave semantics, since a cause is not guaranteed to belong to some $\preceq$-repair. However, if we consider combined complexity, we see that the IAR and brave semantics still retain some computational advantage over AR semantics. This lower complexity comes from the fact that the IAR and brave semantics require only that the answer is entailed by one set of facts (either the intersection of the repairs or some repair), while the AR semantics requires it is entailed by every repair. Note that for the IAR semantics, one can precompute the intersection of repairs in an offline phase and then utilize standard querying algorithms at query time. Finally, if we compare the different types of preferred repairs, we find that the $\subseteq_P$ preorder leads to the lowest complexity, and $\leq_P$ and $\leq_w$ the greatest. However, under the reasonable assumption that there is a bound on the number of priority classes (resp. maximal weight), we obtain the same complexity for the semantics based on $\leq$-, $\leq_P$- and $\leq_w$-repairs.

We should point out that the only properties of DL-Lite$_\mathcal{R}$ that are used in the upper bound proofs are (i) consistency and instance checking are in P w.r.t. combined complexity, and (ii) CQ entailment is in P w.r.t. data complexity and in NP w.r.t. combined complexity. Consequently, our combined complexity upper bounds apply to all ontology languages having polynomial combined complexity for consistency and instance checking and NP combined complexity for CQ entailment, and in particular to the OWL 2 EL profile [Motik *et al.* 2012]. Our data complexity upper bounds apply to all data-tractable ontology languages, which includes Horn DLs [Hustadt *et al.* 2007, Eiter *et al.* 2008] and several classes of the existential rules framework, also called Datalog +/- [Calì *et al.* 2012]. Such classes have been introduced in [Baget *et al.* 2011a, Baget *et al.* 2011b, Krötzsch & Rudolph 2011, Thomazo 2013b].

6.3 Query answering via reduction to SAT for $\subseteq_P$-repair based semantics

Since the $\subseteq_P$-repair based semantics are coNP or NP-complete w.r.t. data complexity, we propose to reduce query answering to SAT, as we did for plain AR semantics.

In the case of the $\subseteq_P$-AR semantics, the most obvious encoding would stipulate that the subset corresponding to a valuation (i) contains no cause for q, (ii) is maximal w.r.t. $\subseteq_P$, and (iii) contains no conflicts. However, such an encoding would contain as many variables as ABox facts, even though most of the ABox may be irrelevant for answering the query at hand. In order to identify potentially relevant assertions, we introduce the notion of an *oriented* conflict graph. In what follows, we use $\alpha \preceq_P \beta$ to signify that there exist $i \leq j$ such that $\alpha \in P_i$ and $\beta \in P_j$, i.e. α has priority over β.

Definition 6.3.1. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_\mathcal{R}$ KB and P be a prioritization of $\mathcal{A}$. The *oriented conflict graph for $\mathcal{K}$ and P*, denoted $G^P_\mathcal{K}$, is the directed graph whose set of vertices is $\mathcal{A}$ and which has an edge from β to α whenever $\alpha \preceq_P \beta$ and $\{\alpha, \beta\}$ is a conflict for $\mathcal{K}$.

The outgoing edges of an assertion in the oriented conflict graph link it to the assertions of same or greater priority which conflict it. The assertions without outgoing edges conflict only assertions of lower priority.

We now give a succinct encoding inspired from that for plain AR semantics, which can be seen as selecting a set of assertions which contradict all of the query's causes, and verifying that this set can be extended to a $\subseteq_P$-repair. Importantly, to check the latter, it suffices to consider only those assertions that are reachable in $G^P_\mathcal{K}$ from an assertion that contradicts some cause.

Theorem 6.3.2. *Let q be a Boolean CQ , $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_\mathcal{R}$ KB, and $P = \langle P_1, \ldots, P_n \rangle$ be a prioritization of $\mathcal{A}$. Consider the following propositional formulas having* *variables of the form x_α for $\alpha \in \mathcal{A}$:*

$$\varphi_{\neg q} = \bigwedge_{\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})} \Big(\bigvee_{\alpha \in \mathcal{C}} \bigvee_{\substack{\beta \in \mathsf{confl}(\{\alpha\},\mathcal{K}) \\ \beta \preceq_P \alpha}} x_\beta \Big)$$

$$\varphi_{\mathsf{max}} = \bigwedge_{\alpha \in R_q} \Big(x_\alpha \vee \bigvee_{\substack{\beta \in \mathsf{confl}(\{\alpha\},\mathcal{K}) \\ \beta \preceq_P \alpha}} x_\beta \Big)$$

$$\varphi_{\mathsf{cons}} = \bigwedge_{\substack{\alpha,\beta \in R_q \\ \beta \in \mathsf{confl}(\{\alpha\},\mathcal{K})}} \neg x_\alpha \vee \neg x_\beta$$

where R_q is the set of assertions reachable in $G^P_\mathcal{K}$ from some assertion β such that x_β appears in $\varphi_{\neg q}$. Then $\mathcal{K} \models_{\subseteq_P\text{-}AR} q$ iff $\varphi_{\neg q} \wedge \varphi_{\mathsf{max}} \wedge \varphi_{\mathsf{cons}}$ is unsatisfiable.

In order to prove Theorem 6.3.2, we begin by establishing the following lemmas that relate the $\subseteq_P$-repairs of $\mathcal{A}$ with the $\subseteq_P$-repairs of R_q.

Lemma 6.3.3. *Every $\subseteq_P$-repair of R_q can be extended to a $\subseteq_P$-repair of $\mathcal{A}$.*

Proof. Let $\mathcal{B}$ be a $\subseteq_P$-repair of R_q. Construct a set $\mathcal{A}'$ by adding to $\mathcal{B}$ a maximal subset $\mathcal{C}_1$ of assertions from $P_1 \setminus \mathcal{B}$ such that $\mathcal{B} \cup \mathcal{C}_1$ is $\mathcal{T}$-consistent, then adding a maximal subset $\mathcal{C}_2$ of assertions from $P_2 \setminus \mathcal{B}$ such that $\mathcal{B} \cup \mathcal{C}_1 \cup \mathcal{C}_2$ is $\mathcal{T}$-consistent, and so on. By construction, the set $\mathcal{A}'$ is $\mathcal{T}$-consistent, and we claim that it is in fact a $\subseteq_P$-repair of $\mathcal{A}$. Suppose for a contradiction that this is not the case. Then there must exist another $\mathcal{T}$-consistent set $\mathcal{B}' \subseteq \mathcal{A}$ and some k such that $\mathcal{A}' \cap P_i = \mathcal{B}' \cap P_i$ for every $1 \leq i < k$, and $\mathcal{A}' \cap P_k \subsetneq \mathcal{B}' \cap P_k$. Consider some $\alpha \in (\mathcal{B}' \cap P_k) \setminus (\mathcal{A}' \cap P_k)$. It follows from the construction of $\mathcal{A}'$ that $\mathcal{B} \cup (\mathcal{A}' \cap (P_1 \cup \ldots \cup P_k)) \cup \{\alpha\}$ is $\mathcal{T}$-inconsistent. Since $(\mathcal{A}' \cap (P_1 \cup \ldots \cup P_k)) \cup \{\alpha\}$ is a subset of $\mathcal{B}'$, $\mathcal{B}'$ is known to be $\mathcal{T}$-consistent, and all conflicts involve at most two assertions, it must be the case that α conflicts with some $\beta \in \mathcal{B} \setminus (P_1 \cup \ldots \cup P_k)$. Since $\beta \in \mathcal{B}$, it must belong to R_q. It follows then from the definition of R_q and the fact that $\alpha \in P_k$ that the assertion α must also belong to R_q. Now consider the set $\mathcal{B}' = (\mathcal{B} \cap (P_1 \cup \ldots \cup P_k)) \cup \{\alpha\}$. It can be easily verified that $\mathcal{B}'$ is a $\mathcal{T}$-consistent set with $\mathcal{B} \subsetneq_P \mathcal{B}'$, contradicting our assumption that $\mathcal{B}$ is a $\subseteq_P$-repair of R_q. □

Lemma 6.3.4. *If $\mathcal{A}'$ is a $\subseteq_P$-repair of $\mathcal{A}$, then $\mathcal{A}' \cap R_q$ is a $\subseteq_P$-repair of R_q.*

Proof. Let $\mathcal{A}'$ be a $\subseteq_P$-repair of $\mathcal{A}$, and set $\mathcal{R} = \mathcal{A}' \cap R_q$. Clearly, $\mathcal{R}$ is $\mathcal{T}$-consistent. Suppose for a contradiction that there exists a set $\mathcal{R}' \subseteq R_q$ such that $\mathcal{R} \subsetneq_P \mathcal{R}'$, and let k be such that $\mathcal{R} \cap P_i = \mathcal{R}'_i \cap P_i$ for all $1 \leq i < k$ and $\mathcal{R} \cap P_k \subsetneq \mathcal{R}' \cap P_k$. We claim that

$$\mathcal{A}'' = ((\mathcal{A}' \setminus R_q) \cap (P_1 \cup \ldots \cup P_k)) \cup \mathcal{R}'$$

satisfies the following:

1. $\mathcal{A}''$ is $\mathcal{T}$-consistent.
2. $\mathcal{A}' \subsetneq_P \mathcal{A}''$.

Note that these statements together contradict our earlier assumption that $\mathcal{A}'$ is a $\subseteq_P$-repair.

To show the first statement, suppose for a contradiction that $\mathcal{A}''$ is $\mathcal{T}$-inconsistent. Since $\mathcal{A}' \setminus R_q$ and $\mathcal{R}'$ are both known to be $\mathcal{T}$-consistent, and conflicts in DL-Lite$_\mathcal{R}$ involve at most two assertions, there must exist a conflict $\{\alpha, \beta\}$ with $\alpha \in (\mathcal{A}' \setminus R_q) \cap (P_1 \cup \ldots \cup P_k)$ and $\beta \in \mathcal{R}'$. Moreover, since $\beta \in R_q$ and $\alpha \notin R_q$, we must have $\beta \prec_P \alpha$. The assertion α belongs to $P_1 \cup \ldots \cup P_k$, so we must have $\beta \in P_j$ for some $j < k$. Since $\mathcal{R} \cap P_i = \mathcal{R}' \cap P_i$ for all $1 \leq i < k$, it follows that $\beta \in \mathcal{R}$, hence $\{\alpha, \beta\} \subseteq \mathcal{A}'$. This is a contradiction, since $\mathcal{A}'$ was assumed to be a $\subseteq_P$-repair, and so must be $\mathcal{T}$-consistent.

For the second statement, we simply note that since $\mathcal{R} \cap P_i = \mathcal{R}'_i \cap P_i$ for all $1 \leq i < k$, we have $\mathcal{A}' \cap P_i = \mathcal{A}'' \cap P_i$ for every $1 \leq i < k$, and since $\mathcal{R} \cap P_k \subsetneq \mathcal{R}' \cap P_k$, we also have $\mathcal{A}' \cap P_k \subsetneq \mathcal{A}'' \cap P_k$. □

Proof of Theorem 6.3.2. We observe that the set of assertions α whose corresponding variable x_α appears in the formula $\varphi_{\neg q} \wedge \varphi_{\mathsf{max}} \wedge \varphi_{\mathsf{cons}}$ is precisely the set R_q. Moreover, every variable x_α with $\alpha \in R_q$ appears in the subformula φ_{max}.

For the first direction, suppose that the formula $\varphi_{\neg q} \wedge \varphi_{\mathsf{max}} \wedge \varphi_{\mathsf{cons}}$ is satisfiable, and let ν be a satisfying truth assignment. Consider the corresponding set of assertions $\mathcal{R}_\nu \subseteq R_q$ consisting of all those assertions α whose corresponding variable x_α is assigned to true by ν. As ν satisfies φ_{cons}, the set $\mathcal{R}_\nu$ contains no conflicts, i.e. it is $\mathcal{T}$-consistent. We claim that $\mathcal{R}_\nu$ is a $\subseteq_P$-repair of R_q. Suppose that this is not the case, and let $\mathcal{R}'$ be a $\mathcal{T}$-consistent subset of R_q such that $\mathcal{R} \cap P_i = \mathcal{R}'_i \cap P_i$ for all $1 \leq i < k$ and $\mathcal{R} \cap P_k \subsetneq \mathcal{R}' \cap P_k$. Consider some $\alpha \in (\mathcal{R}' \setminus \mathcal{R}) \cap P_k$. Since $\alpha \notin \mathcal{R}$, we must have $\nu(x_\alpha) = \mathsf{false}$. As φ_{max} is satisfied by ν, there must exist some variable x_β with $\nu(x_\beta) = \mathsf{true}$ such that the corresponding assertion β satisfies $\beta \in \mathsf{confl}(\{\alpha\}, \mathcal{K})$ and $\beta \preceq_P \alpha$. However, we know that $\mathcal{R} \cap P_i \subseteq \mathcal{R}'_i \cap P_i$ for every $1 \leq i \leq k$, hence $\beta \in \mathcal{R}'$, contradicting the supposed consistency of $\mathcal{R}'$. We have thus shown that $\mathcal{R}_\nu$ is a $\subseteq_P$-repair of R_q. Applying Lemma 6.3.3, we can find a $\subseteq_P$-repair $\mathcal{A}'$ of $\mathcal{A}$ such that $\mathcal{R}_\nu \subseteq \mathcal{A}'$. To show that $\langle \mathcal{T}, \mathcal{A}' \rangle \not\models q$, consider some cause $\mathcal{C}$ for q in $\mathcal{K}$. Then since ν satisfies $\varphi_{\neg q}$, there must exist some assertion $\alpha \in \mathcal{C}$ and some $\beta \in \mathsf{confl}(\{\alpha\}, \mathcal{K})$ such that $\nu(x_\beta) = \mathsf{true}$. It follows that $\beta \in \mathcal{R}$, hence $\beta \in \mathcal{A}'$ and $\mathcal{C} \not\subseteq \mathcal{A}'$. We have thus showed that $\mathcal{A}'$ contains no cause for q. We can thus conclude that $\mathcal{K} \not\models_{\subseteq_P\text{-AR}} q$.

For the other direction, suppose that $\mathcal{K} \not\models_{\subseteq_P\text{-AR}} q$, and let $\mathcal{R}$ be a $\subseteq_P$-repair of $\mathcal{A}$ such that $\langle \mathcal{T}, \mathcal{R} \rangle \not\models q$. Consider the set $\mathcal{R}' = \mathcal{R} \cap R_q$. By Lemma 6.3.4, we have that $\mathcal{R}'$ is a $\subseteq_P$-repair of R_q. Let $\nu_{\mathcal{R}'}$ be the truth assignment that assigns to true precisely those variables x_α for which $\alpha \in \mathcal{R}'$. We wish to show that $\nu_{\mathcal{R}'}$ satisfies $\varphi_{\neg q} \wedge \varphi_{\mathsf{max}} \wedge \varphi_{\mathsf{cons}}$. First, consider some cause $\mathcal{C} \subseteq \mathcal{A}$ for q w.r.t. $\mathcal{K}$. Since $\langle \mathcal{T}, \mathcal{R} \rangle \not\models q$, there must exist an assertion $\alpha \in \mathcal{C}$ that does not appear in $\mathcal{R}$. We also know that $\mathcal{R}$ is a $\subseteq_P$-repair, so there must exist some $\beta \in \mathcal{R}$ with $\beta \preceq_P \alpha$ that conflicts with α (otherwise, we could obtain a more preferred subset by adding α to $\mathcal{R}$ and removing any assertions conflicting with α). The assertion β belongs to R_q, so the variable x_β will be assigned to true by $\nu_{\mathcal{R}'}$, and the clause in $\varphi_{\neg q}$ that corresponds to cause $\mathcal{C}$ is satisfied by $\nu_{\mathcal{R}'}$. We have thus shown that every clause in $\varphi_{\neg q}$ is satisfied.

Next, consider an assertion $\alpha \in R_q$ and its associated clause $x_\alpha \vee \bigvee_{\beta \in \mathsf{confl}(\{\alpha\}, \mathcal{K}), \beta \preceq_P \alpha} x_\beta$ in the formula φ_{max}. If $\alpha \in \mathcal{R}'$, then x_α will be assigned true by $\nu_{\mathcal{R}'}$, and the clause is satisfied. If instead $\alpha \notin \mathcal{R}'$, then also $\alpha \notin \mathcal{R}$. Using the fact that $\mathcal{R}$ is a $\subseteq_P$-repair of $\mathcal{A}$ and similar arguments to above, we can infer that there is some there must exist some $\beta \in \mathcal{R}$ with $\beta \preceq_P \alpha$ and $\beta \in \mathsf{confl}(\{\alpha\},\mathcal{K})$. Since $\alpha \in R_q$, it follows from the definition of the set R_q that $\beta \in R_q$, hence $\beta \in \mathcal{R}'$. We thus have $\nu_{\mathcal{R}'}(x_\beta) = \mathsf{true}$, and so the clause for α is satisfied. This proves that φ_{max} is satisfied by $\nu_{\mathcal{R}'}$.

Finally, since $\mathcal{R}'$ is $\mathcal{T}$-consistent, it contains no conflicts, and so $\nu_{\mathcal{R}'}$ satisfies φ_{cons}. We have thus exhibited a satisfying assignment for the formula $\varphi_{\neg q} \wedge \varphi_{\mathsf{max}} \wedge \varphi_{\mathsf{cons}}$. □

For the $\subseteq_P$-IAR semantics, a query is not entailed just in the case that every cause is absent from some $\subseteq_P$-repair. This can be tested by using one SAT instance per cause.

Theorem 6.3.5. *For each $\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})$, consider the formulas:*

$$\varphi_{\neg\mathcal{C}} = \bigvee_{\alpha\in\mathcal{C}} \bigvee_{\substack{\beta\in\mathsf{confl}(\{\alpha\},\mathcal{K})\\ \beta\preceq_P\alpha}} x_\beta$$

$$\varphi^{\mathcal{C}}_{\mathsf{max}} = \bigwedge_{\alpha\in R_{\mathcal{C}}} (x_\alpha \vee \bigvee_{\substack{\beta\in\mathsf{confl}(\{\alpha\},\mathcal{K})\\ \beta\preceq_P\alpha}} x_\beta)$$

$$\varphi^{\mathcal{C}}_{\mathsf{cons}} = \bigwedge_{\substack{\alpha,\beta\in R_{\mathcal{C}}\\ \beta\in\mathsf{confl}(\{\alpha\},\mathcal{K})}} \neg x_\alpha \vee \neg x_\beta$$

where $R_{\mathcal{C}}$ is the set of assertions reachable in $G^P_{\mathcal{K}}$ from some assertion β such that x_β appears in $\varphi_{\neg\mathcal{C}}$. Then $\mathcal{K} \models_{\subseteq_P\text{-IAR}} q$ iff there exists $\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})$ such that the formula $\varphi_{\neg\mathcal{C}} \wedge \varphi^{\mathcal{C}}_{\mathsf{max}} \wedge \varphi^{\mathcal{C}}_{\mathsf{cons}}$ is unsatisfiable.

Proof. Using similar arguments to the proof of Theorem 6.3.2, one can show that $\varphi_{\neg\mathcal{C}} \wedge \varphi^{\mathcal{C}}_{\mathsf{cons}} \wedge \varphi^{\mathcal{C}}_{\mathsf{max}}$ is satisfiable if and only if there exists a $\subseteq_P$-repair of $\mathcal{A}$ which does not contain the cause $\mathcal{C}$. It follows that $\varphi_{\neg\mathcal{C}} \wedge \varphi^{\mathcal{C}}_{\mathsf{cons}} \wedge \varphi^{\mathcal{C}}_{\mathsf{max}}$ is satisfiable for every $\mathcal{C} \in \mathsf{causes}(q,\mathcal{K})$ just in the case that there is no cause of q in the intersection of the $\subseteq_P$-repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$, i.e. $\mathcal{K} \not\models_{\subseteq_P\text{-IAR}} q$. □

For the $\subseteq_P$-brave semantics, a query is entailed just in the case that some cause is present in some $\subseteq_P$-repair.

Theorem 6.3.6. *Let q be a Boolean CQ, $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ be a DL-Lite$_\mathcal{R}$ KB, and $P = \langle P_1,\ldots,P_n\rangle$ be a prioritization of $\mathcal{A}$.*

$$\varphi_q = (\bigvee_{\mathcal{C}\in\mathsf{causes}(q,\mathcal{K})} w_{\mathcal{C}}) \wedge \bigwedge_{\mathcal{C}\in\mathsf{causes}(q,\mathcal{K})} \bigwedge_{\alpha\in\mathcal{C}} \neg w_{\mathcal{C}} \vee x_\alpha$$

$$\varphi_{\mathsf{max}} = \bigwedge_{\alpha\in R_q} (x_\alpha \vee \bigvee_{\substack{\beta\in\mathsf{confl}(\{\alpha\},\mathcal{K})\\ \beta\preceq_P\alpha}} x_\beta)$$

$$\varphi_{\mathsf{cons}} = \bigwedge_{\substack{\alpha,\beta\in R_q\\ \beta\in\mathsf{confl}(\{\alpha\},\mathcal{K})}} \neg x_\alpha \vee \neg x_\beta$$

where R_q is the set of assertions reachable in $G^P_\mathcal{K}$ from some assertion α such that x_α appears in φ_q. Then $\mathcal{K} \models_{\subseteq_P\text{-brave}} q$ iff $\varphi_q \wedge \varphi_{\mathsf{max}} \wedge \varphi_{\mathsf{cons}}$ is satisfiable.

Proof. As in proof of Theorem 6.3.2, the set of assertions whose corresponding variables are assigned to true in a valuation that satisfies $\varphi_{\mathsf{max}} \wedge \varphi_{\mathsf{cons}}$ can be extended to a $\subseteq_P$-repair. If the valuation also satisfies φ_q, this set of assertion contains a cause for q, so $\mathcal{K} \models_{\subseteq_P\text{-brave}} q$.

In the other direction, if there exists a $\subseteq_P$-repair that contains a cause for q, we can find a valuation that satisfies $\varphi_q \wedge \varphi_{\mathsf{max}} \wedge \varphi_{\mathsf{cons}}$. As in proof of Theorem 6.3.2, if $\mathcal{R}$ is a $\subseteq_P$-repair that contains a cause $\mathcal{C}$ for q, $\mathcal{R}' = \mathcal{R} \cap R_q$ is a $\subseteq_P$-repair of R_q. The valuation $\nu_{\mathcal{R}'}$ that assigns to true the variables whose corresponding assertions are in $\mathcal{R}'$ and assigns $w_\mathcal{C}$ to true and $w'_\mathcal{C}$ to false for every other cause $\mathcal{C}'$ for q satisfies $\varphi_q \wedge \varphi_{\mathsf{max}} \wedge \varphi_{\mathsf{cons}}$. Indeed, since $\mathcal{C} \subseteq \mathcal{R}$ and every assertion of $\mathcal{C}$ has its corresponding variable that appears in φ_q, $\mathcal{C} \subseteq \mathcal{R}'$ so for every $\alpha \in \mathcal{C}$, $\nu_{\mathcal{R}'}(\alpha) = \mathsf{true}$. □

6.4 Implementation and experiments

6.4.1 Consistent query answering with priorities in CQAPri

We implemented query answering under $\subseteq_P$-AR and $\subseteq_P$-IAR semantics within CQAPri. To take into account priorities over ABox assertions, we store the conflicts of the KB as an *oriented* conflict graph, so that each assertion is linked only to its conflicts of same or greater priority.

When a query arrives, CQAPri computes the candidate answers and their images as in the case without priorities, then discards the non-brave answers by checking if all their images contain some conflict, and identifies *an approximation* of the $\subseteq_P$-IAR ones by checking whether there is some image whose assertions have no outgoing edges in the oriented conflict graph. For those (plain) brave-answers that are not found in the approximation of $\subseteq_P$-IAR answers, CQAPri uses the SAT encoding from the preceding section to decide if they hold under $\subseteq_P$-AR semantics. Finally, for the $\subseteq_P$-AR answers not in the approximation of $\subseteq_P$-IAR, it can decide if they hold under $\subseteq_P$-IAR semantics with the SAT encodings for $\subseteq_P$-IAR.

We also implemented an alternative method that consists in first computing the set of assertions that appear in some image of the candidate answer and are in the intersection of the $\subseteq_P$-repairs. For each relevant assertion, we use our encoding to decide if it is entailed under $\subseteq_P$-IAR semantics. We then ignore the outgoing edges of such assertions in the oriented conflict graph when computing the approximation of $\subseteq_P$-IAR answers, which is therefore exactly the set of $\subseteq_P$-IAR answers.

Note that we can observe some high-level similarities between our system and that presented in [Du *et al.* 2013] for querying $\mathcal{SHIQ}$ KBs under $\leq_w$-AR semantics, which also employs SAT solvers and uses a reachability analysis to identify a query-relevant portion of the KB.

6.4.2 Prioritized ABoxes

We put different prioritizations over our benchmark ABoxes. Prioritizations of an ABox were made either by choosing the *same* priority level for *all* the assertions of a concept/role or by choosing *a* priority level for *each* ABox assertion. We built prioritizations this way to capture a variety of scenarios. For instance, a database administrator may (manually) partition an ABox using a few priority levels set by concept/role, based upon the reliability of the business processes that provide the data; an ABox integrating data from many sources may be partitioned with more priority levels set per assertion, with the priority of an assertion depending on the reliability of the sources from which it originates.

We built 4 prioritizations for each ABox, further denoted by the id uXcY of the ABox it derives from, and a suffix pZW first indicating the number Z of priority levels, and then whether priority levels were chosen per concept/role (W = cr) or assertion (W = a). In our experiments, the number of priority levels is 3 and 10.

6.4.3 Experimental results

We measure query answering time and how it varies w.r.t. the size and ratio of conflicts of the ABox depending on the prioritization (3 or 10 levels of priority, set per concept/role or assertion). We also compare our two approaches to see in which cases computing the set of $\subseteq_P$-IAR assertions involved in the problem improves the performances.

The main conclusion of our experiments is that adding priorities really complicates query answering under AR, and even more IAR semantics. In particular, query answering is less robust to the number of conflicts. CQAPri runs out of memory or time (time-out fixed to 3 hours) for many queries on ABoxes which are large or have many assertions in conflicts (e.g. , even on u20c20p3a, CQAPri runs out of time for three queries, and for six when we try to compute the exact set of $\subseteq_P$-IAR answers). However, query answering with priorities remains feasible on realistic cases, when there are only a few assertions in conflict (uXc1pZW case), and on small ABoxes (up to u20cYpZW). We therefore present the results on these cases.

Figures 6.6 and 6.7 show the evolution of the query answering time w.r.t. the proportion of conflicting assertions for u1cYpZW and u20cYpZW in five cases: standard AR query answering, and $\subseteq_P$-AR query answering for the four prioritizations. Figure 6.8 shows the evolution of the query answering time w.r.t. the size of the ABox for the same five cases.

Using priorities makes query answering more sensitive to conflicts because the size of the encodings is related to the number of assertions reachable from the causes in the oriented conflict graph. We observed that using 3 priority levels typically led to harder instances than using 10 levels. Indeed, when there is a large number of priority levels, there are less pairs of conflicting assertions of the same priority, so fewer conflicts translate into bidirectional edges in the oriented conflict graph and the encodings involve fewer assertions. By contrast, it is not clear whether choosing to set priority levels per assertion or per concept and role has an impact on the difficulty of query answering.

Fig. 6.6 Time in seconds for query answering under $\subseteq_P$-AR w.r.t. the ratios of conflicts on u1cYpZW (about 76K assertions), for four prioritizations. For readability, the figures on the bottom focus on the queries whose answering times are lower and whose behaviors are thus not visible on the first one.

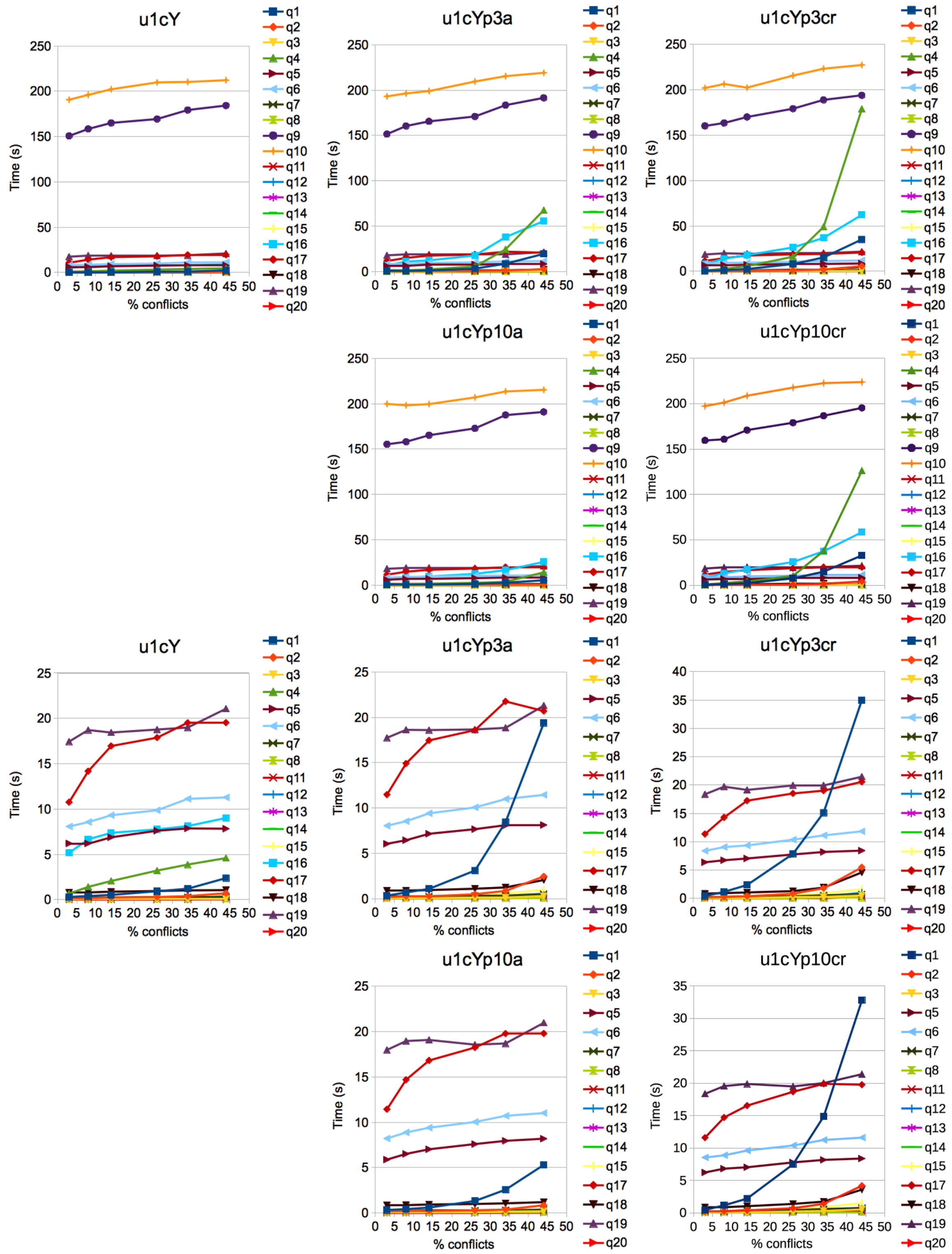


Fig. 6.7 Time in seconds for query answering under $\subseteq_P$-AR w.r.t. the ratios of conflicts on u20cYpZW (about 2 million assertions), for four prioritizations. For readability, the figures on the bottom focus on the queries whose answering times are lower and whose behaviors are thus not visible on the first one.

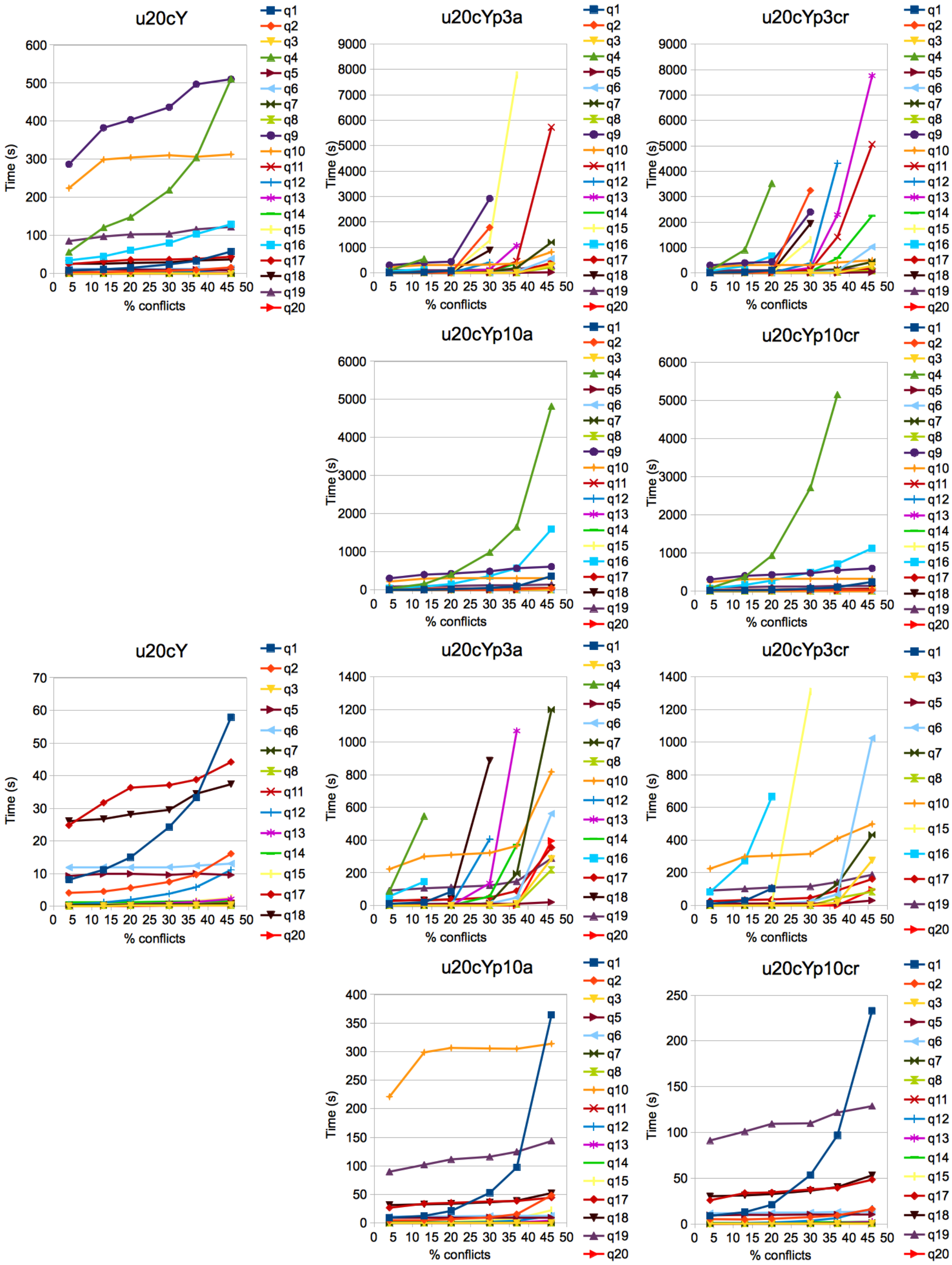


Fig. 6.8 Time in seconds for query answering under $\subseteq_P$-AR w.r.t. the size of the ABox on uXc1pZW (about 4% of assertions involved in some conflict), for four prioritizations. For readability, the figures on the bottom focus on the queries whose answering times are lower and whose behaviors are thus not visible on the first one.

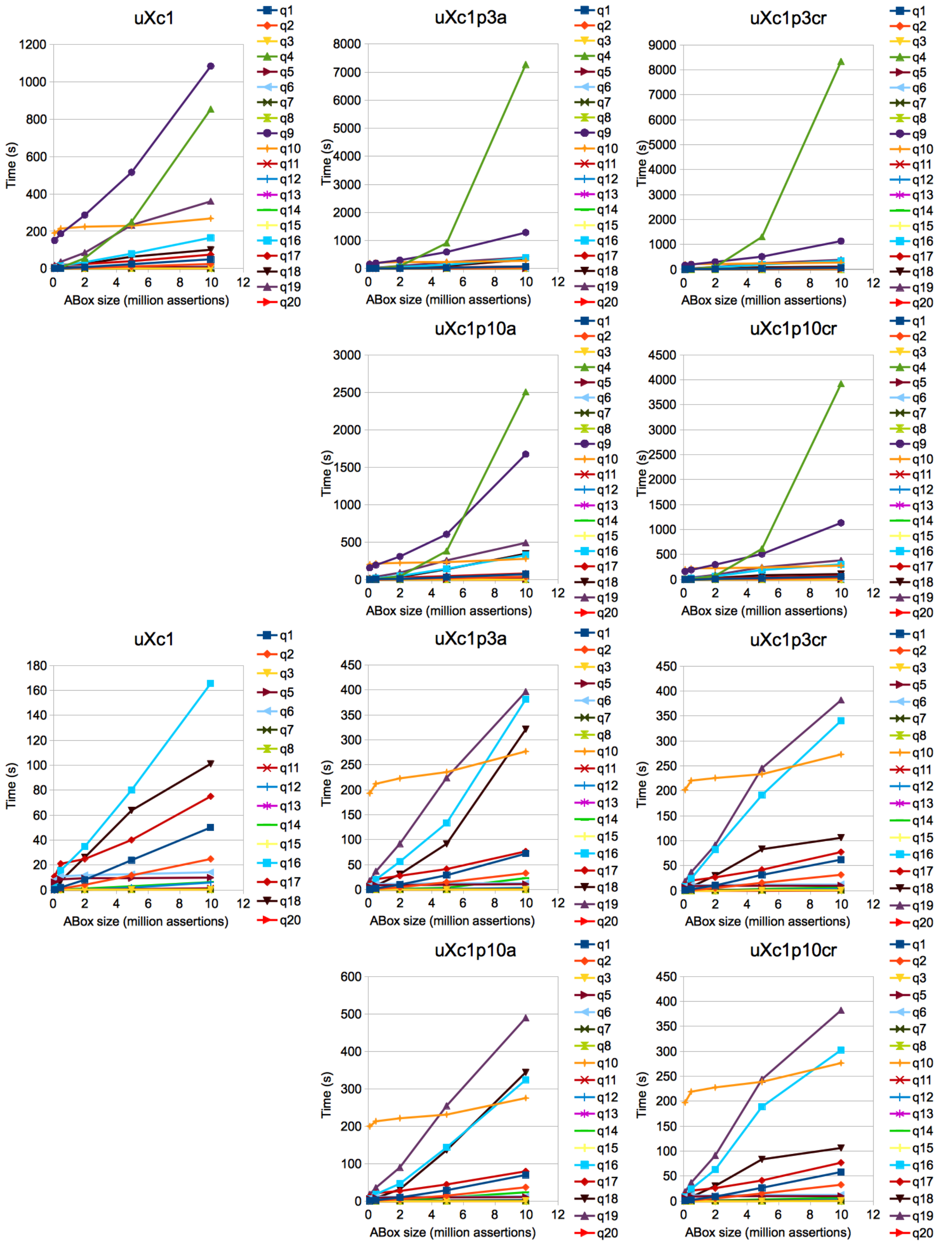


Figure 6.9 shows the evolution of the query answering time w.r.t. the proportion of conflicting assertions for `u1cYp3a` in three cases: $\subseteq_P$-AR query answering, and $\subseteq_P$-AR and $\subseteq_P$-IAR with two different methods: method 1 is the standard method described in Section 6.3, and method 2 consists in deciding for the relevant facts whether they hold under $\subseteq_P$-IAR before deciding whether the answer is entailed under $\subseteq_P$-IAR or $\subseteq_P$-AR semantics. Figure 6.10 shows the evolution of the query answering time w.r.t. the ABox size for `uXc1p3a` in the same cases. Table 6.2 shows, per query for some prioritized ABoxes, how many answers were in the approximation of $\subseteq_P$-IAR, the number of $\subseteq_P$-IAR answers (not in the approximation), of $\subseteq_P$-AR answers (not $\subseteq_P$-IAR), and the number of candidate answers which are not $\subseteq_P$-AR.

Computing the exact set of $\subseteq_P$-IAR answers is very costly when there are lots of $\subseteq_P$-AR answers not in the approximation of $\subseteq_P$-IAR (columns $\subseteq_P$-IAR and $\subseteq_P$-AR in Table 6.2).

The method consisting in computing the set of $\subseteq_P$-IAR assertions among the relevant ones is generally less robust to conflicts and ABox size. However, it sometimes performs better on easy cases.

Fig. 6.9 Time in seconds for query answering under $\subseteq_P$-AR, or $\subseteq_P$-AR and $\subseteq_P$-IAR w.r.t. the proportion of conflicts on u1cYp3a. For readability, the figures on the bottom focus on the queries whose answering times are lower and whose behaviors are thus not visible on the first one.

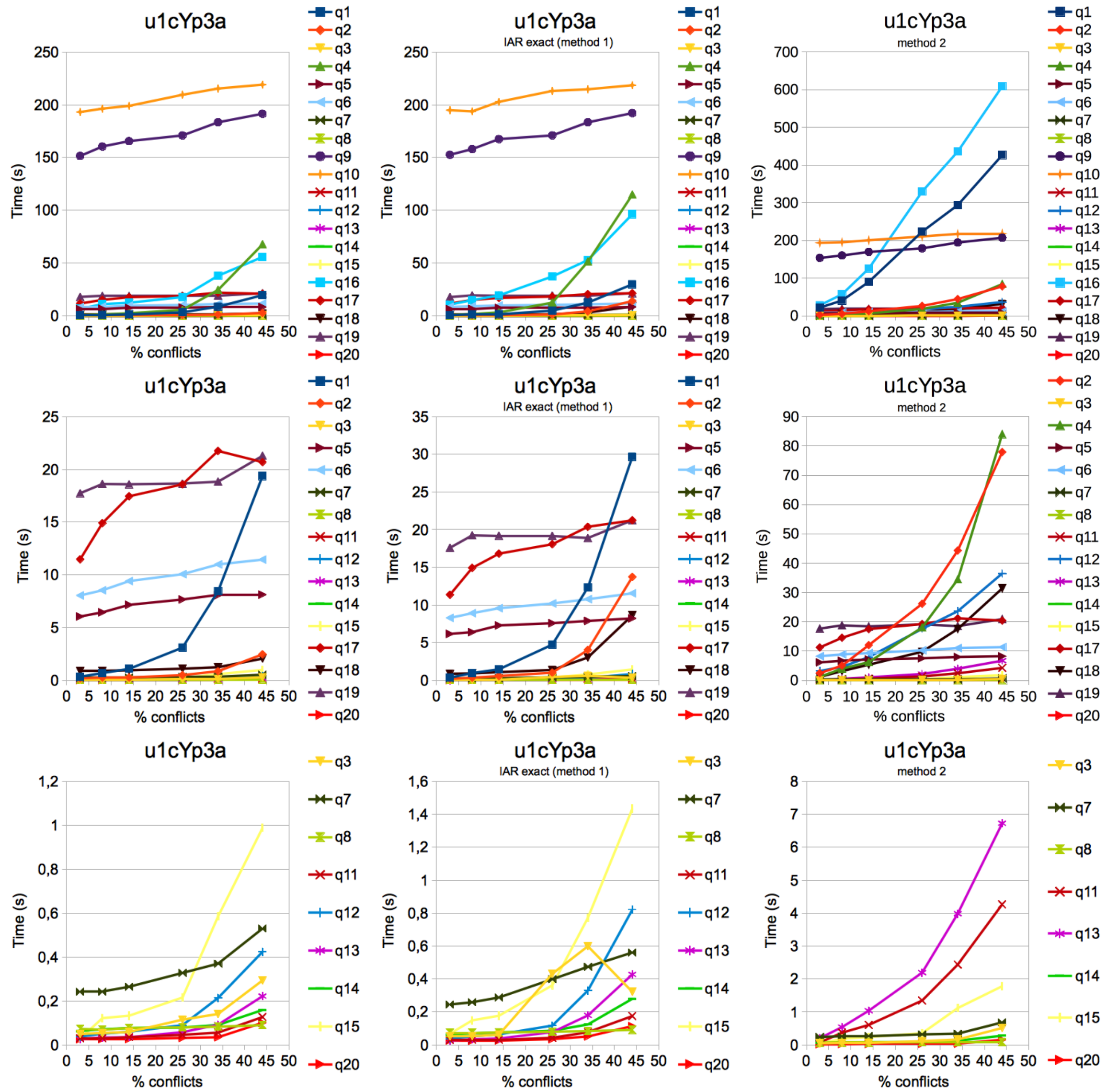


Fig. 6.10 Time in seconds for query answering under $\subseteq_P$-AR, or $\subseteq_P$-AR and $\subseteq_P$-IAR w.r.t. ABox size on uXc1p3a. For readability, the figures on the bottom focus on the queries whose answering times are lower and whose behaviors are thus not visible on the first one.

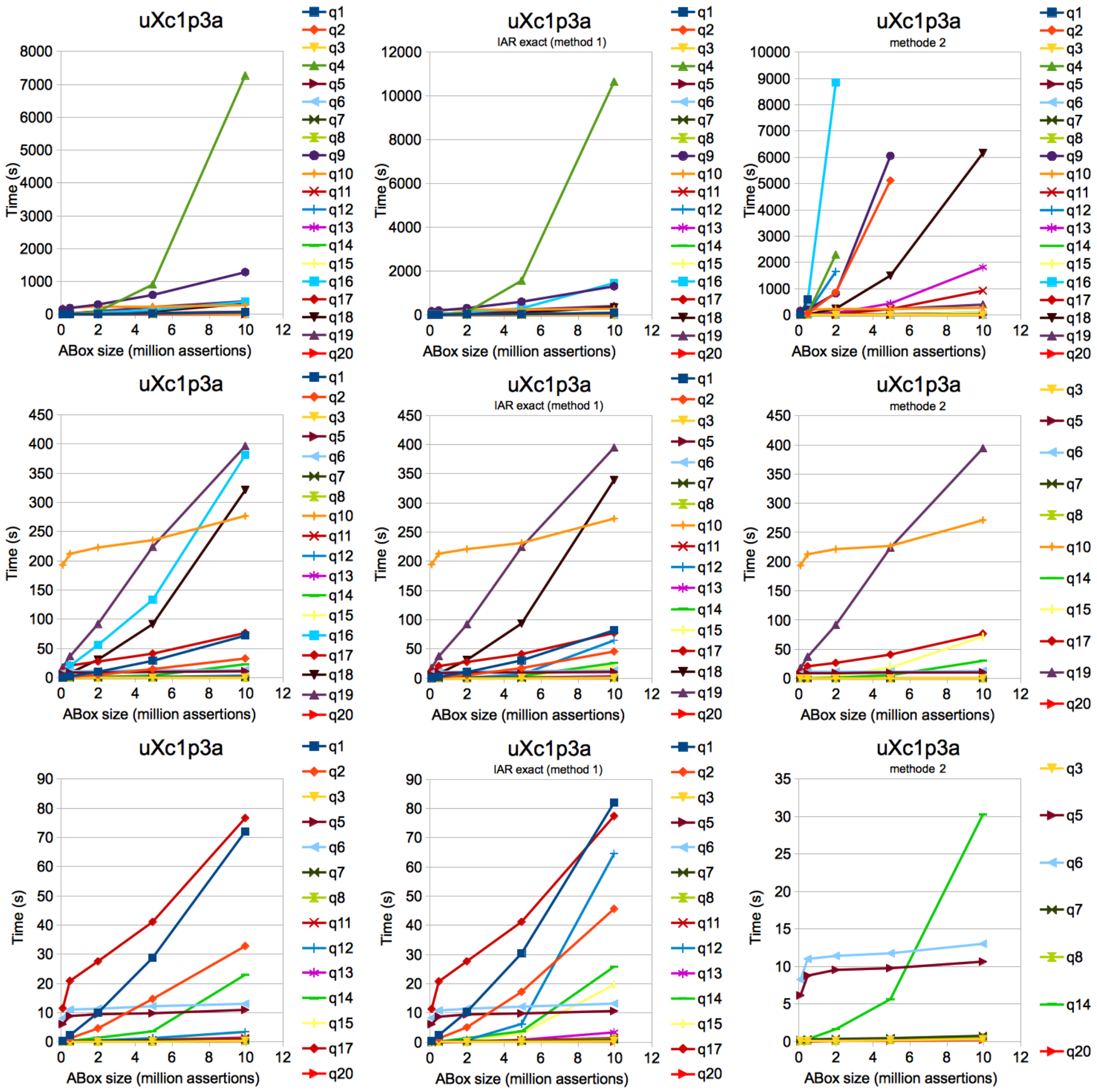


Table 6.2 Number of answers that are in the approximation of $\subseteq_P$-IAR, of $\subseteq_P$-IAR answers (not in the approximation), of $\subseteq_P$-AR answers (not $\subseteq_P$-IAR), and of candidate answers not $\subseteq_P$-AR.

	u1c1p3a				u1c20p3a				u1c50p3a			
	Approx.	$\subseteq_P$-IAR	$\subseteq_P$-AR	Candidate	Approx.	$\subseteq_P$-IAR	$\subseteq_P$-AR	Candidate	Approx.	$\subseteq_P$-IAR	$\subseteq_P$-AR	Candidate
q1	20181	113	0	115	14681	3244	0	2484	9642	5174	0	5596
q2	7220	0	20	7	6812	210	33	365	6039	734	177	752
q3	85	0	0	0	20	48	0	17	0	0	0	87
q4	81052	810	0	1875	35948	19548	0	29285	8324	21627	0	55694
q5	10	0	0	0	10	0	0	0	0	6	0	4
q6	235	0	0	0	207	30	5	59	0	0	0	342
q7	136	1	0	0	29	94	1	14	0	0	0	149
q8	0	0	0	0	0	0	0	0	0	0	0	0
q9	1310	55	2	7	1103	150	17	206	979	154	45	462
q10	2	1	0	0	0	0	0	6	3	0	0	4
q11	537	0	0	1	527	1	0	36	515	8	0	105
q12	1191	1	3	6	1132	3	52	103	1049	6	125	317
q13	1072	1	2	5	1041	20	15	83	997	43	49	214
q14	192	1	0	2	116	50	0	29	55	71	0	69
q15	449	58	0	0	214	155	0	139	50	166	0	299
q16	14201	3218	0	113	3342	7451	0	6739	67	5210	0	12255
q17	0	1	0	0	0	0	0	1	0	0	0	1
q18	3137	1	0	35	2717	137	0	320	2169	425	0	596
q19	0	0	0	0	0	0	0	0	0	0	0	0
q20	50	0	0	0	25	0	0	25	16	8	0	26

	u20c1p3a				u20c20p3a				u20c50p3a			
	Approx.	$\subseteq_P$-IAR	$\subseteq_P$-AR	Candidate	Approx.	$\subseteq_P$-IAR	$\subseteq_P$-AR	Candidate	Approx.	$\subseteq_P$-IAR	$\subseteq_P$-AR	Candidate
q1	536452	5332	0	3323	TO	TO	TO	TO	TO	TO	TO	TO
q2	189557	358	47	471	177253	7609	1868	8946	TO	TO	TO	TO
q3	85	0	0	0	0	0	0	85	11	46	0	30
q4	1375343	388968	0	303475	TO	TO	TO	TO	TO	TO	TO	TO
q5	10	0	0	0	10	0	0	0	4	OOM	6	0
q6	235	0	0	0	180	41	14	66	114	OOM	124	104
q7	105	20	0	12	0	0	0	138	0	0	0	149
q8	21	10	0	0	0	0	0	31	0	0	0	32
q9	33901	1526	17	763	27582	1741	193	9526	TO	TO	TO	TO
q10	41	17	0	0	0	0	0	62	0	0	0	66
q11	14406	4	0	66	14141	63	2	1230	13701	255	15	2998
q12	7305	1	61	184	6451	27	653	3460	TO	TO	TO	TO
q13	28983	28	17	131	28113	508	494	2484	TO	TO	TO	TO
q14	4814	83	0	54	2910	1184	0	857	TO	TO	TO	TO
q15	12666	815	0	271	3626	3910	0	6322	TO	TO	TO	TO
q16	408655	46175	0	13719	TO	TO	TO	TO	TO	TO	TO	TO
q17	27	7	0	3	0	4	0	33	0	0	0	39
q18	82410	224	0	468	70783	4407	0	8063	TO	TO	TO	TO
q19	0	0	0	1	0	0	0	8	1	0	0	19
q20	50	0	0	0	26	0	0	24	0	0	0	50

7

RELATED WORK

In this chapter, we position our work in a more general context and provide more details on some topics mentioned in the other chapters.

7.1 Consistent query answering outside DL

Consistent query answering (CQA), which corresponds to query answering under AR semantics in the database setting, has been introduced in [Arenas *et al.* 1999]. Contrary to DLs, the closed world assumption is generally made, so the facts that are not in the database are considered false. It follows that the violations of integrity constraints of the form $\forall x A(x) \Rightarrow B(x)$ can be solved by removing or adding tuples (e.g. if the database contains only $A(a)$ with the example constraint, we can either remove $A(a)$ or add $B(a)$ to restore consistency). A repair is generally defined as a database that is consistent with the integrity constraints and whose symmetric difference to the original database is inclusion-minimal. The computational complexity of CQA with different kinds of integrity constraints and queries has been extensively studied, and algorithms have been proposed for the different combinations of classes of constraints and queries (see [Bertossi 2006, Chomicki 2007, Bertossi 2011] for surveys about CQA).

In [Calì *et al.* 2003], several semantics are defined for inconsistent and incomplete databases, by considering the database instances that satisfy the integrity constraints and are either supersets (sound semantics) or subsets (complete semantics) of the initial database instance. In order to avoid that the violation of a constraint leads to the non-existence of such instances in the sound case, loose versions of these semantics are introduced. They consider the consistent databases that are "as close as possible" to the actual database instance, w.r.t. the intersection (sound semantics) or the difference (complete semantics) with the initial database. The *loosely-sound semantics* therefore corresponds exactly to the AR semantics, since these database instances correspond to the models of the repairs (in the DL sense) of the actual instance.

Regarding variants of CQA, [Lopatenko & Bertossi 2007] considers repairs whose symmetric difference to the original database is cardinality-maximal, but also aggregate attribute repairs, which are attribute-based repairs (under which database instances can be repaired by changing attributes values in existing tuples only), which minimize a numerical aggregation function over attribute changes throughout the database. Recently, [Pfandler & Sallinger 2015] proposed another variant where the distance between the repairs and the original database is bounded. Note that contrary to preferred repair semantics, a distance-bounded repair is not guaranteed to exist. In this case, every query is considered not entailed under distance-bounded CQA.

Consistent query answering has also been investigated for ontologies given by existential rules. The data complexity of query answering under AR, IAR, and ICR semantics is studied in [Lukasiewicz *et al.* 2013], and [Lukasiewicz *et al.* 2015] analyzes the combined complexity (as well as bounded-arity combined complexity, where the arity of the predicates is bounded, and fixed-program combined complexity, which amounts to fix the TBox in DL setting) of AR query answering for the main decidable classes of existential rules enriched with negative constraints.

Recently, the AR semantics has been generalized in [Eiter *et al.* 2016] so that rules and rule instances, not just database facts, can be removed to restore consistency. This would correspond to removing or altering TBox axioms in the DL setting. *Generalized repairs* are obtained by minimally removing database facts and rules, and *local generalized repairs* by minimally removing database facts and rule instances but not whole rules (e.g. if we have the rules $\forall x A(x) \Rightarrow B(x)$ and $\forall x B(x) \land C(x) \Rightarrow \bot$ and the facts $A(a)$ and $C(a)$, it is possible to restore consistency by removing the rule instance $A(a) \Rightarrow B(a)$). In both cases, it is allowed to define sets of hard rules and facts that cannot be removed.

7.2 Explanations

7.2.1 Justifications of entailed axioms

As mentioned in Chapter 4, there has been significant interest in equipping DL reasoning systems with explanation facilities. The earliest work proposed formal proof systems as a basis for explaining concept subsumptions [McGuinness & Borgida 1995, Borgida *et al.* 2000], while the post-2000 literature mainly focuses on *axiom pinpointing* [Schlobach & Cornet 2003, Kalyanpur *et al.* 2005, Horridge *et al.* 2012], in which the problem is to generate minimal subsets of the KB that yield a given (surprising or undesirable) consequence. Such subsets are often called *justifications*. It should be noted that work on axiom pinpointing has thus far focused on explaining entailed TBox axioms (or possibly ABox assertions), and in particular on TBox debugging by explaining unsatisfiable classes. In our work, we assume that the TBox has been properly debugged, so is consistent and correct, i.e. all consequences of the TBox are desirable. This work on axiom pinpointing can therefore be seen as a first step that allows us to be sure that the errors stem from the data.

For the lightweight DL $\mathcal{EL}+$, justifications have been shown to correspond to minimal models of propositional Horn formulas and can be computed using SAT solvers [Sebastiani & Vescovi 2009]; a polynomial algorithm has been proposed to compute one justification in [Baader *et al.* 2007]. In DL-Lite, the problem is simpler: all justifications can be enumerated in polynomial delay [Peñaloza & Sertkaya 2010].

Beside computing efficiently justifications, several works addressed the problem of making them understandable for the user, either by studying their cognitive complexity [Horridge *et al.* 2011], or by grouping justifications that have a similar structure to help to handle large number of justifications [Bail *et al.* 2013]. Our experiments showed that a query answer can possess a very large number of explanations, many of which are quite similar in structure. It could therefore be interesting to investigate ways of improving the presentation of explanations, e.g. by identifying and grouping similar explanations as has been done for justifications, or by adopting a factorized representation (like in [Olteanu & Zavodny 2012]).

7.2.2 Explanation of query answers

The problem of explaining answers to conjunctive queries over DL-Lite KBs is considered in [Borgida *et al.* 2008] which provides a proof-theoretic approach to explaining positive answers. The proof of an answer involves the ABox assertions and TBox axioms used to derived it. As mentioned in Chapter 4, the difficulty of such proofs could provide an additional criteria for ranking explanations, and the work on the cognitive complexity of justifications may give clues on this difficulty.

Probably the closest related work is [Arioua *et al.* 2015] which introduces an argumentation framework for explaining positive and negative answers under the inconsistency-tolerant semantics ICR. Their motivations are quite similar to our own, and there are some high-level similarities in the definition of explanations (e.g. to explain positive ICR-answers, they consider sets of arguments that minimally cover the preferred extensions, whereas for positive AR-answers, we use sets of causes that minimally cover the repairs). They propose to compute one explanation for a positive or negative ICR-answer with a hitting set algorithm, applied either on the sets of supporting arguments (which correspond to our causes) present in each extension (corresponding to the repair), or on the set of attacking arguments (which correspond to the conflicts of the causes). Our work differs from theirs by considering different semantics and by providing detailed complexity analysis, in which we do not assume that the set of repairs is given, and an implemented prototype. Another argumentation framework has been proposed for ground BCQs explanation under IAR and brave semantics in [Arioua & Croitoru 2016a], then under AR semantics in [Arioua & Croitoru 2016b], and has been implemented in the DALEK prototype.

Finally, we note that the problem of explaining query results has been studied in the database community (cf. [Cheney *et al.* 2009] for a survey). The *lineage* of a query answer is the set of tuples of the database that contribute to produce the answer, i.e. the union of the images for the answer. The *why-provenance* corresponds to the images and the *minimal witness basis* to the set of causes of the answer. The *how-provenance* describes how a result was produced from the tuples. The *where-provenance* provides the location (i.e. relation, tuple and attribute) of the values in the answer tuple.

7.2.3 Query abduction

Since we defined explanations for a negative AR- or IAR-answer using assertions of the ABox that conflict its causes, defining explanations for negative brave-answers (which have no cause) would significantly differ in spirit. The problem of explaining negative answers has been primarily seen as the problem of finding a minimal dataset to be added to the data to get the missing answers, i.e. as *query abduction*, both in the DL (cf. [Calvanese *et al.* 2013, Du *et al.* 2014] for DL-Lite and [Wang *et al.* 2015] for $\mathcal{ELH}_{\bot}$) and in the database arena (cf. [Herschel & Hernández 2010]). In both settings, restrictions on the signature of the explanation are allowed. Note that a different approach related to query debugging was proposed in the database context [Bidoit *et al.* 2014], and focuses on finding subqueries responsible for pruning the missing answer from a query result. This approach is less relevant to our setting since conjunctive queries over DL KBs are much simpler and less error-prone than SQL queries.

The complexity of the decision problems related to explaining negative answers in DL-Lite (recognition and existence of an explanation, necessity and relevance of an assertion) is studied in [Calvanese *et al.* 2013]. An implementation that computes explanations in DL-Lite was presented in [Du *et al.* 2014] and the case of inconsistent KBs is treated in [Du *et al.* 2015], where an explanation is a set of assertions to add that will lead to the answer holding under IAR semantics. These three papers tackle the issue of preferred explanations: Calvanese et al. consider subset or cardinality-minimal explanations, whereas Du et al. introduce the notion of representative explanation, which is an explanation that is minimal (when allowing renaming of fresh individuals to compare explanations) and is not subsumed by any other (e.g. if $\mathsf{Advise}(ann, ann)$, $\mathsf{Advise}(ann, bob)$, and $\mathsf{Advise}(ann, ind)$ where ind is a fresh individual are the explanations for ann not being an answer to $\exists y \mathsf{Advise}(x, y)$, the last one is the unique representative explanation), and cardinality-minimal preferred explanations for a preference relation based on cardinality-preserving substitutions.

The idea of representative explanation could be used for presenting our explanations, treating the individuals that are mapped to existentially quantified variables in the same way as Du et al. treat fresh individuals. The framework of query abduction could also be a useful starting point for providing users with suggestions of assertions to add when insertions are needed to satisfy some answers during query-driven repairing. As mentioned in Chapter 5, it would be natural to restrict the signature of the explanations to avoid adding some predicates. It would also be crucial to exploit the interaction between the answers of the QRP and to define proper preference relations. For instance, we should favor the insertion of assertions that do not create causes for unwanted answers or participate in creating causes for several unsatisfied wanted answers.

7.3 Evolution, revision and updates

The problem of making a knowledge base evolve has been intensively explored in the past decades. Belief revision [Gärdenfors 1992] consists in incorporating new information in a KB while preserving consistency. Different operators have been defined for removing or adding a sentence to a logical theory. A commonly agreed criterion is that the changes to the original belief set have to be minimal, but minimality can be defined at the syntactic level (*formula-based changes*), or at the level of the models of the knowledge base (*model-based changes*). The former approach seems more reasonable if the KB corresponds to a body of explicit belief, while the latter respects the principle of irrelevance of syntax that expresses that two KBs which have the same models are equivalent so should lead to equivalent revised KBs. Note that in our setting, since we treat inconsistent KBs that have no models, formula-based approaches make more sense.

When there are several KBs that realize the revision and differ minimally from the initial KB, several strategies have been proposed. Among the approaches considered in [Eiter & Gottlob 1992], the three formula-based approaches consider the formulas true in all (Ginsberg's approach), some (Cross-Product), or the intersection (When In Doubt Throw It Out) of the possible worlds. The AR, brave and IAR semantics correspond to these approaches. The latter is used in [Nebel 1991] in the context of belief revision, and [Fagin *et al.* 1983] considers the Cross-Product strategy for updating databases.

It is widely accepted that some pieces of knowledge are more important or reliable than others. Partitioning the KB into levels of priority is a natural solution [Fagin *et al.* 1983, Nebel 1991]. In particular, in [Fagin *et al.* 1983], they are used to make database integrity constraints take priority over database facts, since the authors consider that the constraints should not be given up because of an update in the data that violates them.

Many works address belief change in the DL setting. The model-based approach for instance-level update and erasure (i.e. addition or deletion of a set of assertions) is studied in [De Giacomo *et al.* 2009], with a focus on DL-Lite$_{\mathcal{F}}$. Since the KB resulting from such an update may not be expressible in the DL of the initial KB, the authors introduce a minimal extension of DL-Lite$_{\mathcal{F}}$ closed for instance-level updates and use it to construct an approximated instance-level update and erasure for DL-Lite$_{\mathcal{F}}$ KB. The formula-based approach is considered in [Calvanese *et al.* 2010] which proposes novel update strategies for which evolution is expressible in DL-Lite to overcome the limitation of the Cross-Product approach (which leads to KBs not expressible in DL-Lite), while losing less information than with the "When In Doubt Throw It Out" approach. Similar problems have been explored in [Calvanese *et al.* 2015], where different inconsistency-tolerant semantics for querying and updating knowledge and action bases are defined.

While the new piece of information is traditionally considered more reliable than the initial KB, [Ahmeti *et al.* 2016] defines three semantics for SPARQL instance-level updates that do not systematically give preference to the update: the cautious semantics rejects updates potentially introducing conflicts; the brave semantics gives favor to the new information, and the fainthearted semantics is a compromise between the former two approaches which incorporates as much of the new information as possible, as long as consistency with the prior knowledge is not violated.

Closer to our work on query-driven repairing is the problem of modifying DL KBs to ensure (non) entailments of assertions. The erasure operation, which consists in deleting an entailed statement, is tackled from a practical point of view in [Gutierrez *et al.* 2011] for RDFS knowledge bases. When the erasure concerns the schema, the possible solutions correspond to the minimal cuts in the path from the causes to the unwanted consequence. In the case of instance erasure, when the schema should be left untouched, the erasure is unique and amounts to removing the (singleton) causes of the unwanted assertion.

Our query-driven repairing framework is inspired by that of [Jiménez-Ruiz *et al.* 2011], in which a user specifies two sets of axioms that should be entailed or not by a KB. Repair plans are introduced as pairs of sets of axioms to remove and add to obtain an ontology satisfying these requirements. Deletion-only repair plans are studied in [Jiménez-Ruiz *et al.* 2009] where heuristics based on the confidence and the size of the plan are used to help the user to choose a plan among all minimal plans. Axioms that occur in all plans (i.e. are necessary to remove to be able to find a solution) are also marked. We adapted their framework to the setting of conjunctive queries and inconsistent KBs, using inconsistency-tolerant semantics. This leads us to consider not only the causes of the answers, but also the conflicts of the causes of wanted answers, since a tuple may be a negative IAR-answer because of the presence of erroneous assertions that contradict its causes. The main difference with our work is in the treatment of the problem: Jiménez-Ruiz et al. compute all solutions (if there are any) and present them to the user, while we investigate the case in which the assertions that can be added are not in a known finite set (so it is not possible to compute all solutions), and we look for approximations when there is no solution and ensure that a solution will actually satisfy every answer. In the case of deletion-only repair plans, instead of computing all solutions and presenting them, we propose to help the user to find which assertions to remove. We integrate a notion of impact similar to that presented in [Nikitina *et al.* 2012]. The latter paper studies the problem of interactive KB revision where each axiom of a KB has to be (in)validated by an expert, and the goal is to reduce the expert effort by ordering the axioms to evaluate to maximize the automatic (in)validation of axioms.

Compared with the work of Jiménez-Ruiz et al., one distinguishing feature of our work is the specifications of the changes at the level of query answers. The main idea of our query-driven repairing approach is the same as in [Bergman *et al.* 2015]: erroneous or missing answers are indicated by the user and the data has to be repaired by asking the user if some tuples are valid or not. The strategy presented in the latter work consists in repeating the two following steps until the objectives are achieved: (i) remove all erroneous answers, one after the other, by removing singleton causes (necessarily false) then questioning the user about the tuples that appear in the highest number of causes; (ii) add the missing answers by asking the user to complete a partial cause.The authors also study the impact of using several imperfect experts for this process and try to split the query to help the user to find a cause for a missing answer. The main difference of our approach is that we take into account all answers at the same time to increase the number of necessarily false assertions. The number of causes in which a tuple appears, as well as other criteria mentioned in [Bergman *et al.* 2015], such as responsibility or trust scores could be considered to refine our ranking.

Note that the problem of modifying the data to make some consequences hold is also related to the problem of view update in databases where the data modification is specified at the level of a view, which corresponds to a query, and the corresponding database update has to be found [Fagin *et al.* 1983].

7.4 Inconsistency and uncertainty handling in DL

The problem of dealing with inconsistencies has been considered in several areas of knowledge representation and reasoning (cf. survey [Bertossi *et al.* 2005]). Regarding DL-based ontologies, [Haase *et al.* 2005] surveys four different approaches to handling inconsistency: consistent ontology evolution, which has been discussed in the preceding section, locating and repairing inconsistencies, reasoning in the presence of inconsistencies, and multi-version reasoning, which considers not only the latest version of an ontology, but also all previous versions to deal with inconsistencies that arise from the interaction of the ontology with its environment.

A general framework for reasoning in the presence of inconsistent ontologies is presented in [Huang *et al.* 2005]. The authors define some desirable properties for an inconsistent reasoner, among them soundness, which requires that every formula that follows from an inconsistent KB follows from a consistent subset using a classical reasoner, and meaningfulness, which requires that the reasoner does not allow to derive contradictory statements. In our setting, semantics that are sound approximations of brave fulfill the soundness criteria, and complete approximations of AR, as well as CAR and ICAR and k-lazy semantics are meaningful according to this definition.

An alternative way to get meaningful answers from an inconsistent theory is to adopt *paraconsistent logics*, which have more than two truth values (most often four (Belnap's logic): true, false, undefined and over-defined). For DL KBs, a four-valued semantics is defined in [Zhou *et al.* 2012, Maier *et al.* 2013]. We give the idea of this semantics using simple ontologies consisting of axioms of the forms $A \sqsubseteq B$ and $A \sqsubseteq \neg B$, where $A, B \in \mathsf{N_C}$. A four-valued interpretation $\mathcal{I}$ over a domain $\Delta^{\mathcal{I}}$ maps each concept A to a pair of sets $\langle A_P^{\mathcal{I}}, A_N^{\mathcal{I}} \rangle$ of individuals. The inclusion $A \sqsubseteq B$ is satisfied if $A_P^{\mathcal{I}} \subseteq B_P^{\mathcal{I}}$, $A \sqsubseteq \neg B$ is satisfied if $A_P^{\mathcal{I}} \subseteq B_N^{\mathcal{I}}$, and the assertion $A(a)$ is satisfied if $a^{\mathcal{I}} \in A_P^{\mathcal{I}}$. Models and entailment are defined as usual. Under this semantics, the KB $\langle \{A \sqsubseteq \neg B\}, \{A(a), B(a)\} \rangle$ has models, like $A_P^{\mathcal{I}} = \{a^{\mathcal{I}}\}$, $A_N^{\mathcal{I}} = \emptyset$ and $B_P^{\mathcal{I}} = \{a^{\mathcal{I}}\}$, $B_N^{\mathcal{I}} = \{a^{\mathcal{I}}\}$, and both $A(a)$ and $B(a)$ are entailed. Note that $A \sqsubseteq \neg B$ and $B \sqsubseteq \neg A$ are no longer equivalent. This approach is orthogonal to inconsistency-tolerant semantics. If a DL allows us to use intersection and bottom, it is possible to construct a KB that has no models in the four-valued semantics, like $\langle \{A \sqcap \neg B \sqsubseteq \bot\}, \{A(a), B(a)\} \rangle$, while inconsistency-tolerant semantics always give meaningful results.

Inconsistency is quite related to uncertainty. First because uncertainty may arise from inconsistency, for instance the answers to a query under different inconsistency-tolerant semantics could be seen as more or less certain. Second because inconsistency can be avoided to some extent by expressing that we are not sure of some pieces of knowledge, so that two contradictory but unsure statements can coexist. Different ways of dealing with uncertain DL KBs have been proposed.

The first and most widely studied proposition is the use of probability to model uncertainty. There has been a number of proposals for *probabilistic DLs*. In [Jaeger 1994], probabilistic KBs associate probabilities to the ABox assertions, and the TBox contains both classical terminological axioms and probabilistic terminological axioms that take the form of conditional probability which express statistical probabilities. In [Lutz & Schröder 2010], the authors introduce an operator that allows them to construct complex concepts involving priorities and to express probability on conjunctions of (negated) assertions. In the OMQA setting, [Jung & Lutz 2012] introduces probabilistic ABoxes that associate expressions built on probabilistic events to ABox assertions and probabilities to these events, and computes the probabilities of conjunctive query answers.

In *possibilistic DLs* [Qi *et al.* 2007], each axiom is associated with a degree of certainty, and the α-cut of a KB consists of the axioms of degree greater than α. The inconsistency degree Inc of a KB is the maximal degree such that the corresponding cut is inconsistent, and the plausible consequences of the KB are the consequences of its Inc-cut. Recently, [Benferhat *et al.* 2015] investigated inconsistency-tolerant semantics in possibilistic DL-Lite, which extend the plausible consequences, for instance by adding to the Inc-cut the assertions that are free of conflicts, or conflicted only by assertions of lower degree of certainty.

Subjective DLs [Garcia *et al.* 2015] extend ABox assertions with opinions (b, d, u) where b is a degree of belief, d the degree of disbelief, and u the degree of uncertainty ($b + d + u = 1$). The semantics is given in terms of interpretations that map each individual to an element of the domain, and each concept (resp. role) to a function that associates to every element (resp. pair of elements) of the domain an opinion. For instance, an interpretation $\mathcal{I}$ satisfies an assertion $A(a)$ if the degree of belief and disbelief associated to $A^{\mathcal{I}}(a^{\mathcal{I}})$ are greater than those of the opinion associated to $A(a)$. The problem of instance checking amounts to find the most general opinion associated to a given assertion, and query answering is defined accordingly.

Note that *fuzzy DLs* [Straccia 2001] model vagueness rather than uncertainty, by providing degree of membership for concepts for which there exists no sharp distinction between members and nonmembers (e.g. for two concepts describing temperature cold and hot, some intermediate values could be assigned to both concepts, with a degree of membership lower than 1. Notice that in this case, it would not make sense to state that these two concepts are disjoint).

8

Conclusion and perspectives

Summary of our contributions

In this thesis, we defended the idea that it is possible to deal with inconsistent data in a satisfying and efficient way, which is a crucial point for spreading the use of ontologies allowing for disjointness axioms or functional roles. Indeed, until now, most of ontology-based data-rich applications simply avoid inconsistency, either by using simple ontologies that do not allow for contradiction, or by rejecting any update of the data that will lead to inconsistency. We focused on the AR semantics, which is the most well-known and arguably the most natural semantics, since it can be seen as a generalization to the inconsistent case of the classical semantics, under which the certain answers have to hold in every model of the KB. We also use the IAR and brave semantics, which we consider as lower and upper bounds of the possible answers. As a first step, we proposed an efficient approach to compute the answers under these three semantics, and our experimental evaluation showed promising results. We then addressed three problems that should support the adoption of this framework: *query result explanation*, to help the user to understand the different levels of confidence of the answers he obtained, *query-driven repairing*, to capitalize on the user feedback about query results to improve the data quality, and *preferred repair semantics*, to take into account the reliability of the ABox assertions. For these three issues, we came up with a solution framework, analyzed the complexity of the related problems, and proposed and implemented algorithms, which we empirically studied over an inconsistent DL-Lite$_\mathcal{R}$ benchmark we built. Our results indicate that even if the problems related to dealing with inconsistent DL-Lite$_\mathcal{R}$ KBs are theoretically hard, they can often be solved efficiently in practice by using tractable approximations and features of modern SAT solvers.

Discussion and perspectives

We conclude this dissertation by discussing some of our results and possible directions for future work.

Going beyond DL-Lite$_{\mathcal{R}}$

Throughout the thesis, we focused on DL-Lite$_{\mathcal{R}}$ for simplicity and because it is the basis for the W3C standard OWL 2 QL. It is natural to wonder what happens for other ontology languages. Most of our results actually hold for other dialects of the DL-Lite family, but the problem of reasoning with the inconsistency-tolerant semantics we consider becomes considerably harder for many other well-known DLs.

Extension of our results to other languages of the DL-Lite family We recall that DL-Lite$_{\mathsf{core}}$ is the core language of the family and amounts to DL-Lite$_{\mathcal{R}}$ without role inclusions. DL-Lite$_{\mathcal{F}}$ extends DL-Lite$_{\mathsf{core}}$ with functionality axioms on roles or on their inverses of the form $(\mathsf{funct}\ S)$. DL-Lite$_{\mathcal{A}}$ extends DL-Lite$_{\mathsf{core}}$ with both role inclusions and functionality with the restriction that functional roles cannot be specialized, i.e. used positively on the right-hand side of a role inclusion.

The complexity of query answering, consistency checking and computation of the causes for a query and the conflicts of a knowledge base are the same for all these languages, and the size of the conflicts is at most two in all cases. Therefore, all complexity upper bounds presented in the thesis hold for DL-Lite$_{\mathsf{core}}$, DL-Lite$_{\mathcal{R}}$, DL-Lite$_{\mathcal{F}}$ and DL-Lite$_{\mathcal{A}}$. Moreover, since all our algorithms only need the causes and conflicts, they can be used without any modification. To make our prototype CQAPri able to handle KBs expressed in DL-Lite$_{\mathcal{F}}$ or DL-Lite$_{\mathcal{A}}$, we simply need to modify the computation of the conflicts of the knowledge base that currently only search for violation of disjointness TBox axioms in order to also find pairs of assertions that contradict functionality axioms.

Regarding complexity lower bounds, all hardness results of Chapters 4 and 6 hold for KBs expressed in DL-Lite$_{\mathsf{core}}$, hence for these four DL-Lite dialects. Indeed, the reductions used to prove hardness in these chapters do not use role inclusions. In Chapter 5, the complexity results for potential solutions (Theorem 5.3.5) hold for KBs expressed in DL-Lite$_{\mathsf{core}}$, but the reductions used in the proofs of the complexity results related to optimal repair plans (Theorem 5.2.16) use role inclusions. Note that if we allow the restriction of the signature of the assertions that can be added, the hardness results hold for DL-Lite$_{\mathsf{core}}$: we can avoid using role inclusions in the reductions if we restrict this signature. For the complexity of recognizing an optimal repair plan with AR (Theorem 5.4.10), the Π^p_2-hardness results hold for DL-Lite$_{\mathsf{core}}$, but not the $\Delta^p_2[O(\log \mathrm{n})]$-hardness results. The latter hold in the case where the insertion signature is restricted.

Regarding Horn versions of DL-Lite, which allow the use of conjunctions in the concepts appearing in the left-hand side of TBox inclusions, our complexity results do not hold because in this case the size of the causes is not bounded by q, so they cannot be computed in P.

DL languages outside the DL-Lite family When moving outside the DL-Lite family, query answering under the AR, or even the IAR or brave semantics becomes hard [Rosati 2011]. For $\mathcal{ALC}$, which is the prototypical expressive description logic, query answering under these three semantics is in the second level of the polynomial hierarchy w.r.t. data complexity, even for instance queries, while it is coNP-complete under classical semantics. Even for $\mathcal{EL}_\bot$, which extends the lightweight DL $\mathcal{EL}$ with the ability to express disjointness using $\bot$, query answering under IAR and brave semantics is intractable w.r.t. data complexity for instance queries, while query answering is in P under classical semantics. However, for $\mathcal{EL}_\bot$, AR query answering has the same complexity as in DL-Lite. It would be interesting to see if query answering under these semantics can be done efficiently for $\mathcal{EL}_\bot$, since the IAR semantics does not provide a tractable approximation of AR in this setting.

Using other query answering techniques

Our approach is based on the computation of the causes of the answers, which are obtained straightforwardly using the UCQ rewritings of the initial queries. Our system uses the tool Rapid that implements the algorithm of [Chortaras *et al.* 2011] to reformulate the query, but we could use any algorithm that produces such rewritings. However, database management systems perform poorly on large UCQs, and it is not uncommon for UCQ rewritings to be (very) large (there were more than 200,000 CQs in the rewritings of some queries of our benchmark!). That is why FOL rewritings that can be evaluated more efficiently in practice than the standard UCQs have been proposed. In [Thomazo 2013a], an algorithm that rewrites a conjunctive query in a union of semi-conjunctive queries (which are conjunctions of disjunctions) is proposed. In [Bursztyn *et al.* 2015, Bursztyn *et al.* 2016], the authors introduce a space of reformulated queries which are joins of unions of conjunctive queries, from which the reformulated query with the lowest estimated evaluation cost is selected. Another promising approach is the use of non-recursive Datalog rewritings. The algorithm Presto presented in [Rosati & Almatelli 2010] generates a non-recursive Datalog program instead of a UCQ, and produces a query that is not exponential in the number of atoms of the initial query but only in the number of a subset of the join variables of the query typically much smaller. In [Eiter *et al.* 2012], the Clipper system, which implements an algorithm for rewriting queries over Horn-$\mathcal{SHIQ}$ ontologies that transforms the query into a Datalog program ready for evaluation over any ABox, performs well over DL-Lite KBs.

To improve the performance of our system for queries that have a long evaluation time, we could try to use such optimized techniques to find the candidate answers, and then evaluate only the Boolean UCQ rewritings instantiated with these answers to retrieve their causes. Indeed, they would contain less variables and so would be easier to evaluate than the original UCQs. It would probably be useful for queries with a limited number of answers. An open question is whether it is possible to compute the causes more directly, without the UCQs.

Note that the alternative method to rewriting for query answering in DL-Lite, namely the combined approach [Lutz *et al.* 2013], which saturates the data by adding to the ABox every assertion that can be derived and introduces constants to witness existential role restrictions, then uses a special rewriting to prune spurious answers, cannot be straightforwardly adapted to our setting. Indeed, this approach is based on the fact that a DL-Lite KB has a canonical model which is such that the answers of a query in the canonical model are the same as those over the KB, and the inconsistency-tolerant semantics are based on repairs that each has its own canonical model. For IAR semantics, it is possible to compute the intersection of the repairs then use the combined approach over the resulting consistent KB, but in this case updating the saturated intersection when the original ABox is modified will be non-trivial.

Developing and enhancing this work

Explanation framework The explanations we defined provide the basic information needed to understand positive and negative query answers under the AR, IAR and brave semantics. We could build a more complete explanation framework which allows the user to ask for the justification of such explanations (e.g. why is this set of assertions a cause for the query? which causes are contradicted by this assertion and why? in which causes is this assertion?). Such justifications would involve both ABox and TBox axioms and rely on related work about justifications of entailed axioms and query answer explanations. They could be computed on demand and we could find clues on how to select or present them in the literature (cf. Chapter 7.2).

Query-driven repairing framework Our work on query-driven repairing gives the basis for partially cleaning the data based on the user feedback at query time. There are many possible improvements to our algorithms. First, when insertions are allowed and needed, it is very important to help the user to find what insert. Besides the work on query abduction that will be very useful, we could use the specific aspects of this setting to propose insertions. For instance, when the wanted answers were initially entailed under brave semantics and lost their causes because of the deletion of some assertion, we could suggest that the user adds an assertion that “replaces” the assertion deleted (e.g. if we removed $\mathsf{AProf}(ann)$ and lose the wanted answer $\exists x \mathsf{Prof}(x)$, we could suggest to add $\mathsf{Prof}(ann)$). In the same vein, we could present the consequences of the deleted assertions to the user to ask him if he wants to preserve some of them. It would also be interesting to see how allowing the insertion of ABox assertions using variables, as it is done for the language introduced in [De Giacomo *et al.* 2009], impacts the problem. Another approach which could be interesting is to try to find which assertions have to be removed by asking more general questions to the user to obtain part of his knowledge. For instance suppose that a QRP involves several assertions from which derives $\mathsf{Prof}(ann)$, and others from which derives $\mathsf{Postdoc}(ann)$, that $\mathsf{Person}(ann)$ is wanted and that Prof and $\mathsf{Postdoc}$ are disjoint subconcepts of Person. Then asking for the truth value of $\mathsf{Prof}(ann)$ will allow us to discover that several assertions are false if the answer is different from $\mathsf{unknown}$. Finally, our algorithm for deletion-only repair plans could be improved by refining the impact we use to rank the assertions, optimizing its computation or considering ranking only a part of the relevant assertions.

Preferred repairs We provided algorithms for the semantics based on $\subseteq_P$, but did not investigate further the three other cases, which are computationally harder. It would be interesting to find a good way of performing query answering under these semantics, especially $\leq_w$, which can also translate different levels of reliability but allow for compensation between the priority levels.

It would also be interesting to extend our explanation framework to the case of preferred repair semantics. Indeed, a starting point for explaining a positive or negative answer under these semantics would be the explanations defined in the same way as in the classical case (i.e. causes included in some preferred repair, or in the intersection of the preferred repairs, disjunctions of causes that cover the preferred repairs, minimal consistent subsets of the

ABox contradicting every cause and belonging to some preferred repair...), but we also need to explain why a given cause is in some or all preferred repairs for instance. In the case of the $\subseteq_P$ based semantics, it should still be possible to provide explanations that involve only a restricted part of the ABox, since it is possible to reason locally to decide whether a query is entailed or not under these semantics.

System, benchmarks and experiments Our CQAPri system is a prototype, which aims at analyzing the behavior of our algorithms. We therefore focused on the algorithmic part of the system rather than on the query evaluation part or on the presentation of query results. A real system for inconsistency-tolerant query answering should be able to use different rewriting systems, be more flexible on the way of storing the data, and implement an interface for efficiently displaying the results and interacting with the user. Working on these points would probably result in gains in performance.

We did our best to experimentally evaluate our algorithms and built an experimental setting over a well-known benchmark by adding contradictions in data in a way as realistic as we could. However, $\text{LUBM}^{\exists}_{20}$ is an artificial benchmark whose data is very regular, and it would be important to find more natural and varied data to test our framework in a more realistic setting. As discussed in Section 3.4.2, there have been very few experiments conducted over DL-Lite KBs with inconsistent ABoxes, and we did not find any satisfying benchmark.

Another parameter that should be taken into account in the experiments to go further is the user. Indeed, when providing explanations or support for repairing, the major objective is to help a user to operate inconsistent data. The next step should therefore be to conduct experiments with users to understand better what are their needs and how the explanations or suggestions of assertions to remove to repair the ABox are useful and can be improved.

Going a step further

We focused on the setting where the TBox is consistent and assumed to be correct. Little work has been done without this assumption, and there are interesting questions there.

Inconsistency-tolerant semantics could also be combined with other approaches for uncertainty handling (probabilistic or possibilistic DLs for instance), or other settings that take into account some other dimensions of the data, like vagueness with fuzzy DLs, or temporal aspects.

REFERENCES

[Ahmeti *et al.* 2016] Albin Ahmeti, Diego Calvanese, Axel Polleres et Vadim Savenkov. *Handling Inconsistencies Due to Class Disjointness in SPARQL Updates*. In Proceedings of ESWC, 2016.

[Altwaijry *et al.* 2013] Hotham Altwaijry, Dmitri V. Kalashnikov et Sharad Mehrotra. *Query-Driven Approach to Entity Resolution*. PVLDB, vol. 6, no. 14, pages 1846–1857, 2013.

[Amgoud & Vesic 2011] Leila Amgoud et Srdjan Vesic. *A new approach for preference-based argumentation frameworks*. Annals of Mathematics and Artificial Intelligence, vol. 63, no. 2, pages 149–183, 2011.

[Arenas *et al.* 1999] Marcelo Arenas, Leopoldo E. Bertossi et Jan Chomicki. *Consistent Query Answers in Inconsistent Databases*. In Proceedings of PODS, 1999.

[Arioua & Croitoru 2016a] Abdallah Arioua et Madalina Croitoru. *Dialectical Characterization of Consistent Query Explanation with Existential Rules*. In Proceedings of FLAIRS, 2016.

[Arioua & Croitoru 2016b] Abdallah Arioua et Madalina Croitoru. *A Dialectical Proof Theory for Universal Acceptance in Coherent Logic-based Argumentation Frameworks*. In Proceedings of ECAI, 2016.

[Arioua *et al.* 2014a] Abdallah Arioua, Nouredine Tamani et Madalina Croitoru. *On Conceptual Graphs and Explanation of Query Answering under Inconsistency*. In Proceedings of ICCS, 2014.

[Arioua *et al.* 2014b] Abdallah Arioua, Nouredine Tamani, Madalina Croitoru et Patrice Buche. *Query Failure Explanation in Inconsistent Knowledge Bases Using Argumentation*. In Proceedings of COMMA, 2014.

[Arioua *et al.* 2015] Abdallah Arioua, Nouredine Tamani et Madalina Croitoru. *Query Answering Explanation in Inconsistent Datalog +/- Knowledge Bases*. In Proceedings of DEXA, 2015.

[Artale *et al.* 2009] Allessandro Artale, Diego Calvanese, Roman Kontchakov et Michael Zakharyaschev. *The DL-Lite Family and Relations*. Journal Artif. Intell. Res. (JAIR), vol. 36, pages 1–69, 2009.

References

[Baader *et al.* 2003] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi et Peter F. Patel-Schneider, editors. The description logic handbook: Theory, implementation and applications. Cambridge University Press, 2003.

[Baader *et al.* 2007] Franz Baader, Rafael Peñaloza et Boontawee Suntisrivaraporn. *Pinpointing in the Description Logic* EL^+. In Proceedings of KI, 2007.

[Baget *et al.* 2011a] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier et Eric Salvat. *On rules with existential variables: Walking the decidability line*. Artif. Intell., vol. 175, no. 9-10, pages 1620–1654, 2011.

[Baget *et al.* 2011b] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph et Michaël Thomazo. *Walking the Complexity Lines for Generalized Guarded Existential Rules*. In Proceedings of IJCAI, 2011.

[Baget *et al.* 2016] Jean-François Baget, Salem Benferhat, Zied Bouraoui, Madalina Croitoru, Marie-Laure Mugnier, Odile Papini, Swan Rocher et Karim Tabia. *A General Modifier-Based Framework for Inconsistency-Tolerant Query Answering*. In Proceedings of KR, 2016.

[Bail *et al.* 2013] Samantha Bail, Bijan Parsia et Ulrike Sattler. *The logical diversity of explanations in OWL ontologies*. In Proceedings of CIKM, 2013.

[Benferhat *et al.* 2015] Salem Benferhat, Zied Bouraoui et Karim Tabia. *Non Defeated-Based Repair in Possibilistic DL-Lite Knowledge Bases*. In Proceedings of FLAIRS, 2015.

[Bergman *et al.* 2015] Moria Bergman, Tova Milo, Slava Novgorodov et Wang-Chiew Tan. *QOCO: A Query Oriented Data Cleaning System with Oracles*. PVLDB, vol. 8, no. 12, pages 1900–1911, 2015.

[Berre & Parrain 2010] Daniel Le Berre et Anne Parrain. *The Sat4j library, release 2.2*. JSAT, vol. 7, no. 2-3, pages 59–64, 2010.

[Bertossi *et al.* 2005] Leopoldo E. Bertossi, Anthony Hunter et Torsten Schaub, editors. Inconsistency Tolerance [result from a Dagstuhl seminar], volume 3300 of *Lecture Notes in Computer Science*. Springer, 2005.

[Bertossi 2006] Leopoldo E. Bertossi. *Consistent query answering in databases*. SIGMOD Record, vol. 35, no. 2, pages 68–76, 2006.

[Bertossi 2011] Leopoldo E. Bertossi. Database repairing and consistent query answering. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[Bidoit *et al.* 2014] Nicole Bidoit, Melanie Herschel et Katerina Tzompanaki. *Query-Based Why-Not Provenance with NedExplain*. In Proceedings of EDBT, 2014.

[Bienvenu & Ortiz 2015] Meghyn Bienvenu et Magdalena Ortiz. *Ontology-Mediated Query Answering with Data-Tractable Description Logics*. In Lecture Notes of the 11th International Reasoning Web Summer School, volume 9203 of *LNCS*, pages 218–307. Springer, 2015.

[Bienvenu & Rosati 2013] Meghyn Bienvenu et Riccardo Rosati. *Tractable Approximations of Consistent Query Answering for Robust Ontology-based Data Access*. In Proceedings of IJCAI, 2013.

[Bienvenu *et al.* 2014] Meghyn Bienvenu, Camille Bourgaux et François Goasdoué. *Querying Inconsistent Description Logic Knowledge Bases under Preferred Repair Semantics*. In Proceedings of AAAI, 2014.

[Bienvenu *et al.* 2016a] Meghyn Bienvenu, Camille Bourgaux et François Goasdoué. *Explaining Inconsistency-Tolerant Query Answering over Description Logic Knowledge Bases*. In Proceedings of AAAI, 2016.

[Bienvenu *et al.* 2016b] Meghyn Bienvenu, Camille Bourgaux et François Goasdoué. *Query-driven Repairing of Inconsistent DL-Lite Knowledge Bases*. In Proceedings of IJCAI, 2016.

[Bienvenu 2012] Meghyn Bienvenu. *On the Complexity of Consistent Query Answering in the Presence of Simple Ontologies*. In Proceedings of AAAI, 2012.

[Borgida *et al.* 2000] Alexander Borgida, Enrico Franconi et Ian Horrocks. *Explaining ALC Subsumption*. In Proceedings of ECAI, 2000.

[Borgida *et al.* 2008] Alexander Borgida, Diego Calvanese et Mariano Rodriguez-Muro. *Explanation in the DL-Lite Family of Description Logics*. In Proceedings of OTM, 2008.

[Brewka *et al.* 2008] Gerhard Brewka, Ilkka Niemelä et Miroslaw Truszczynski. *Preferences and Nonmonotonic Reasoning*. AI Magazine, vol. 29, no. 4, pages 69–78, 2008.

[Bursztyn *et al.* 2015] Damian Bursztyn, François Goasdoué et Ioana Manolescu. *Efficient Query Answering in DL-Lite through FOL Reformulation (Extended Abstract)*. In Proceedings of DL, 2015.

[Bursztyn *et al.* 2016] Damian Bursztyn, François Goasdoué et Ioana Manolescu. *Teaching an RDBMS about ontological constraints*. PVLDB, 2016.

[Buss & Hay 1991] Samuel R. Buss et Louise Hay. *On Truth-Table Reducibility to SAT*. Information and Computation, vol. 91, no. 1, pages 86–102, 1991.

[Calì *et al.* 2003] Andrea Calì, Domenico Lembo et Riccardo Rosati. *On the decidability and complexity of query answering over inconsistent and incomplete databases*. In Proceedings of PODS, 2003.

[Calì *et al.* 2012] Andrea Calì, Georg Gottlob et Thomas Lukasiewicz. *A general Datalog-based framework for tractable query answering over ontologies*. Journal Web Sem., vol. 14, pages 57–83, 2012.

[Calvanese *et al.* 2007] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini et Riccardo Rosati. *Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family*. Journal of Automated Reasoning (JAR), vol. 39, no. 3, pages 385–429, 2007.

[Calvanese *et al.* 2010] Diego Calvanese, Evgeny Kharlamov, Werner Nutt et Dmitriy Zheleznyakov. *Evolution of DL-Lite Knowledge Bases*. In Proceedings of ISWC, 2010.

[Calvanese *et al.* 2011] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi et Domenico Fabio Savo. *The MASTRO system for ontology-based data access*. Semantic Web, vol. 2, no. 1, pages 43–53, 2011.

[Calvanese *et al.* 2013] Diego Calvanese, Magdalena Ortiz, Mantas Simkus et Giorgio Stefanoni. *Reasoning about Explanations for Negative Query Answers in DL-Lite*. Journal of Artificial Intelligence Research (JAIR), vol. 48, pages 635–669, 2013.

[Calvanese *et al.* 2015] Diego Calvanese, Marco Montali et Ario Santoso. *Verification of Generalized Inconsistency-Aware Knowledge and Action Bases*. In Proceedings of IJCAI, 2015.

[Cheney *et al.* 2009] James Cheney, Laura Chiticariu et Wang Chiew Tan. *Provenance in Databases: Why, How, and Where*. Foundations and Trends in Databases, vol. 1, no. 4, pages 379–474, 2009.

[Chomicki *et al.* 2004a] Jan Chomicki, Jerzy Marcinkowski et Slawomir Staworko. *Computing consistent query answers using conflict hypergraphs*. In Proceedings of CIKM, 2004.

[Chomicki *et al.* 2004b] Jan Chomicki, Jerzy Marcinkowski et Slawomir Staworko. *Hippo: A System for Computing Consistent Answers to a Class of SQL Queries*. In Proceedings of EDBT, 2004.

[Chomicki 2007] Jan Chomicki. *Consistent Query Answering: Five Easy Pieces*. In Proceedings of ICDT, 2007.

[Chortaras *et al.* 2011] Alexandros Chortaras, Despoina Trivela et Giorgos Stamou. *Optimized Query Rewriting for OWL 2 QL*. In Proceedings of CADE, 2011.

[De Giacomo *et al.* 2009] Giuseppe De Giacomo, Maurizio Lenzerini, Antonella Poggi et Riccardo Rosati. *On Instance-level Update and Erasure in Description Logic Ontologies*. Journal of Logic and Computation, vol. 19, no. 5, pages 745–770, 2009.

[Du *et al.* 2013] Jianfeng Du, Guilin Qi et Yi-Dong Shen. *Weight-based consistent query answering over inconsistent $\mathcal{SHIQ}$ knowledge bases*. Knowledge and Information Systems, vol. 34, no. 2, pages 335–371, 2013.

[Du *et al.* 2014] Jianfeng Du, Kewen Wang et Yi-Dong Shen. *A Tractable Approach to ABox Abduction over Description Logic Ontologies*. In Proceedings of AAAI, 2014.

[Du *et al.* 2015] Jianfeng Du, Kewen Wang et Yi-Dong Shen. *Towards Tractable and Practical ABox Abduction over Inconsistent Description Logic Ontologies*. In Proceedings of AAAI, 2015.

[Eiter & Gottlob 1992] Thomas Eiter et Georg Gottlob. *On the Complexity of Propositional Knowledge Base Revision, Updates, and Counterfactuals*. Artif. Intell., vol. 57, no. 2-3, pages 227–270, 1992.

[Eiter & Gottlob 1995] Thomas Eiter et Georg Gottlob. *The Complexity of Logic-Based Abduction*. Journal of the ACM, vol. 42, no. 1, pages 3–42, 1995.

[Eiter & Gottlob 1997] Thomas Eiter et Georg Gottlob. *The Complexity Class Θ_2^P: Recent Results and Applications in AI and Modal Logic*. In Proceedings of FCT, 1997.

[Eiter *et al.* 2008] Thomas Eiter, Georg Gottlob, Magdalena Ortiz et Mantas Simkus. *Query Answering in the Description Logic Horn-$\mathcal{SHIQ}$*. In Proceedings of JELIA, 2008.

[Eiter *et al.* 2012] Thomas Eiter, Magdalena Ortiz, Mantas Simkus, Trung-Kien Tran et Guohui Xiao. *Query Rewriting for Horn-SHIQ Plus Rules*. In Proceedings of AAAI, 2012.

[Eiter *et al.* 2016] Thomas Eiter, Thomas Lukasiewicz et Livia Predoiu. *Generalized Consistent Query Answering under Existential Rules*. In Proceedings of KR, 2016.

[Fagin *et al.* 1983] Ronald Fagin, Jeffrey D. Ullman et Moshe Y. Vardi. *On the Semantics of Updates in Databases*. In Proceedings of PODS, 1983.

[Fuxman & Miller 2005] Ariel Fuxman et Renée Journal Miller. *First-Order Query Rewriting for Inconsistent Databases*. In Proceedings of ICDT, 2005.

[Fuxman *et al.* 2005] Ariel Fuxman, Elham Fazli et Renée J. Miller. *ConQuer: Efficient Management of Inconsistent Databases*. In Proceedings of SIGMOD, 2005.

[Garcia *et al.* 2015] Jhonatan Garcia, Jeff Z. Pan et Achille Fokoue. *Handling uncertainty: An extension of DL-Lite with Subjective Logic*. In Proceedings of DL, 2015.

[Gärdenfors 1992] Peter Gärdenfors. Belief Revision. Cambridge University Press, Cambridge, 1992.

[Gottlob *et al.* 2014] Georg Gottlob, Giorgio Orsi et Andreas Pieris. *Query Rewriting and Optimization for Ontological Databases*. ACM Trans. Database Syst., vol. 39, no. 3, page 25, 2014.

[Gottlob 1995] Georg Gottlob. *NP Trees and Carnap's Modal Logic*. Journal of the ACM, vol. 42, no. 2, 1995.

[Guo *et al.* 2005] Yuanbo Guo, Zhengxiang Pan et Jeff Heflin. *LUBM: A benchmark for OWL knowledge base systems*. J. Web Sem., vol. 3, no. 2-3, pages 158–182, 2005.

[Gutierrez *et al.* 2011] Claudio Gutierrez, Carlos A. Hurtado et Alejandro A. Vaisman. *RDFS Update: From Theory to Practice*. In Proceedings of ESWC, 2011.

[Haase *et al.* 2005] Peter Haase, Frank van Harmelen, Zhisheng Huang, Heiner Stuckenschmidt et York Sure. *A Framework for Handling Inconsistency in Changing Ontologies*. In Proceedings of ISWC, 2005.

[Herschel & Hernández 2010] Melanie Herschel et Mauricio A. Hernández. *Explaining Missing Answers to SPJUA Queries*. PVLDB, vol. 3, no. 1, pages 185–196, 2010.

[Horridge *et al.* 2011] Matthew Horridge, Samantha Bail, Bijan Parsia et Ulrike Sattler. *The Cognitive Complexity of OWL Justifications*. In Proceedings of ISWC, 2011.

[Horridge *et al.* 2012] Matthew Horridge, Bijan Parsia et Ulrike Sattler. *Extracting Justifications from BioPortal Ontologies*. In Proceedings of ISWC, 2012.

[Huang *et al.* 2005] Zhisheng Huang, Frank van Harmelen et Annette ten Teije. *Reasoning with Inconsistent Ontologies*. In Proceedings of IJCAI, 2005.

[Hustadt *et al.* 2007] Ullrich Hustadt, Boris Motik et Ulrike Sattler. *Reasoning in Description Logics by a Reduction to Disjunctive Datalog*. Journal Aut. Reasoning (JAR), vol. 39, no. 3, pages 351–384, 2007.

[Jaeger 1994] Manfred Jaeger. *Probabilistic Reasoning in Terminological Logics*. In Proceedings of KR, 1994.

[Jiménez-Ruiz *et al.* 2009] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks et Rafael Berlanga Llavori. *Ontology Integration Using Mappings: Towards Getting the Right Logical Consequences*. In Proceedings of ESWC, 2009.

[Jiménez-Ruiz *et al.* 2011] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks et Rafael Berlanga Llavori. *Supporting concurrent ontology development: Framework, algorithms and tool*. Data & Knowledge Engineering Journal (DKE), vol. 70, no. 1, pages 146–164, 2011.

[Jung & Lutz 2012] Jean Christoph Jung et Carsten Lutz. *Ontology-Based Access to Probabilistic Data with OWL QL*. In Proceedings of ISWC, 2012.

[Kalyanpur *et al.* 2005] Aditya Kalyanpur, Bijan Parsia, Evren Sirin et James A. Hendler. *Debugging unsatisfiable classes in OWL ontologies*. Journal Web Sem., vol. 3, no. 4, pages 268–293, 2005.

[Kolaitis *et al.* 2013] Phokion G. Kolaitis, Enela Pema et Wang-Chiew Tan. *Efficient Querying of Inconsistent Databases with Binary Integer Programming*. PVLDB, vol. 6, no. 6, pages 397–408, 2013.

[Kontokostas *et al.* 2014] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen et Amrapali Zaveri. *Test-driven evaluation of linked data quality*. In Proceedings of WWW, 2014.

[Krentel 1988] Mark W. Krentel. *The Complexity of Optimization Problems*. Journal of Computer and System Sciences, vol. 36, no. 3, pages 490–509, 1988.

[Krötzsch & Rudolph 2011] Markus Krötzsch et Sebastian Rudolph. *Extending Decidable Existential Rules by Joining Acyclicity and Guardedness*. In Proceedings of IJCAI, 2011.

[Lembo *et al.* 2010] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi et Domenico Fabio Savo. *Inconsistency-Tolerant Semantics for Description Logics*. In Proceedings of RR, 2010.

[Lembo *et al.* 2011] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi et Domenico Fabio Savo. *Query Rewriting for Inconsistent DL-Lite Ontologies*. In Proceedings of RR, 2011.

[Lembo *et al.* 2015] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi et Domenico Fabio Savo. *Inconsistency-tolerant query answering in ontology-based data access*. Journal Web Sem., vol. 33, pages 3–29, 2015.

[Liberatore 2005] Paolo Liberatore. *Redundancy in logic I: CNF propositional formulae*. Artif. Intell. (AIJ), 2005.

[Lopatenko & Bertossi 2007] Andrei Lopatenko et Leopoldo E. Bertossi. *Complexity of Consistent Query Answering in Databases Under Cardinality-Based and Incremental Repair Semantics*. In Proceedings of ICDT, 2007.

[Lukasiewicz *et al.* 2012] Thomas Lukasiewicz, Maria Vanina Martinez et Gerardo I. Simari. *Inconsistency Handling in Datalog+/- Ontologies*. In Proceedings of ECAI, 2012.

[Lukasiewicz *et al.* 2013] Thomas Lukasiewicz, Maria Vanina Martinez et Gerardo I. Simari. *Complexity of Inconsistency-Tolerant Query Answering in Datalog+/-*. In Proceedings of OTM, 2013.

[Lukasiewicz *et al.* 2015] Thomas Lukasiewicz, Maria Vanina Martinez, Andreas Pieris et Gerardo I. Simari. *From Classical to Consistent Query Answering under Existential Rules*. In Proceedings of AAAI, 2015.

[Lutz & Schröder 2010] Carsten Lutz et Lutz Schröder. *Probabilistic Description Logics for Subjective Uncertainty*. In Proceedings of KR, 2010.

[Lutz *et al.* 2013] Carsten Lutz, Inanç Seylan, David Toman et Frank Wolter. *The Combined Approach to OBDA: Taming Role Hierarchies Using Filters*. In Proceedings of ISWC, 2013.

[Maier *et al.* 2013] Frederick Maier, Yue Ma et Pascal Hitzler. *Paraconsistent OWL and related logics*. Semantic Web, vol. 4, no. 4, pages 395–427, 2013.

[Marileo & Bertossi 2010] Mónica Caniupán Marileo et Leopoldo E. Bertossi. *The consistency extractor system: Answer set programs for consistent query answering in databases*. Data Knowl. Eng., vol. 69, no. 6, pages 545–572, 2010.

[McGuinness & Borgida 1995] Deborah L. McGuinness et Alexander Borgida. *Explaining Subsumption in Description Logics*. In Proceedings of IJCAI, 1995.

[Meilicke *et al.* 2008] Christian Meilicke, Heiner Stuckenschmidt et Andrei Tamilin. *Supporting Manual Mapping Revision using Logical Reasoning*. In Proceedings of AAAI, 2008.

[Meliou *et al.* 2010] Alexandra Meliou, Wolfgang Gatterbauer, Katherine F. Moore et Dan Suciu. *The Complexity of Causality and Responsibility for Query Answers and non-Answers*. PVLDB, vol. 4, no. 1, pages 34–45, 2010.

[Motik *et al.* 2012] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue et Carsten Lutz. *OWL 2 Web Ontology Language Profiles*. W3C Recommendation, 11 December 2012. Available at http://www.w3.org/TR/owl2-profiles/.

[Nebel 1991] Bernhard Nebel. *Belief Revision and Default Reasoning: Syntax-Based Approaches*. In Proceedings of KR, 1991.

[Nebel 1998] B. Nebel. Handbook of defeasible reasoning and uncertainty management systems, volume 3: Belief change, chapitre How Hard is it to Revise a Belief Base? Kluwer, 1998.

[Nikitina *et al.* 2012] Nadeschda Nikitina, Sebastian Rudolph et Birte Glimm. *Interactive ontology revision*. Journal of Web Semantics (JWS), vol. 12, pages 118–130, 2012.

[Olteanu & Zavodny 2012] Dan Olteanu et Jakub Zavodny. *Factorised representations of query results: size bounds and readability*. In Proceedings of ICDT, 2012.

[Peñaloza & Sertkaya 2010] Rafael Peñaloza et Baris Sertkaya. *Complexity of Axiom Pinpointing in the DL-Lite Family of Description Logics*. In Proceedings of ECAI, 2010.

[Pérez-Urbina *et al.* 2009] Héctor Pérez-Urbina, Ian Horrocks et Boris Motik. *Efficient Query Answering for OWL 2*. In Proceedings of ISWC, 2009.

[Pfandler & Sallinger 2015] Andreas Pfandler et Emanuel Sallinger. *Distance-Bounded Consistent Query Answering*. In Proceedings of IJCAI, 2015.

[Qi *et al.* 2007] Guilin Qi, Jeff Z. Pan et Qiu Ji. *A Possibilistic Extension of Description Logics*. In Proceedings of DL, 2007.

[Rodriguez-Muro *et al.* 2013] Mariano Rodriguez-Muro, Roman Kontchakov et Michael Zakharyaschev. *Ontology-Based Data Access: Ontop of Databases*. In Proceedings of ISWC, 2013.

[Rosati & Almatelli 2010] Riccardo Rosati et Alessandro Almatelli. *Improving Query Answering over DL-Lite Ontologies*. In Proceedings of KR, 2010.

[Rosati *et al.* 2012] Riccardo Rosati, Marco Ruzzi, Mirko Graziosi et Giulia Masotti. *Evaluation of Techniques for Inconsistency Handling in OWL 2 QL Ontologies*. In Proceedings of ISWC, 2012.

[Rosati 2011] Riccardo Rosati. *On the Complexity of Dealing with Inconsistency in Description Logic Ontologies*. In Proceedings of IJCAI, 2011.

[Schlobach & Cornet 2003] Stefan Schlobach et Ronald Cornet. *Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies*. In Proceedings of IJCAI, 2003.

[Sebastiani & Vescovi 2009] Roberto Sebastiani et Michele Vescovi. *Axiom Pinpointing in Lightweight Description Logics via Horn-SAT Encoding and Conflict Analysis*. In Proceedings of CADE, 2009.

[Staworko *et al.* 2012] Slawek Staworko, Jan Chomicki et Jerzy Marcinkowski. *Prioritized repairing and consistent query answering in relational databases*. Annals of Mathematics and Artificial Intelligence, vol. 64, no. 2-3, pages 209–246, 2012.

[Straccia 2001] Umberto Straccia. *Reasoning within Fuzzy Description Logics*. J. Artif. Intell. Res. (JAIR), vol. 14, pages 137–166, 2001.

[Thomazo 2013a] Michaël Thomazo. *Compact Rewritings for Existential Rules*. In Proceedings of IJCAI, 2013.

[Thomazo 2013b] Michaël Thomazo. *Conjunctive Query Answering Under Existential Rules - Decidability, Complexity, and Algorithms*. PhD thesis, Montpellier 2 University, France, 2013.

[Venetis *et al.* 2012] Tassos Venetis, Giorgos Stoilos et Giorgos B. Stamou. *Incremental Query Rewriting for OWL 2 QL*. In Proceedings of DL, 2012.

[Wagner 1987] Klaus W. Wagner. *More Complicated Questions About Maxima and Minima, and Some Closures of NP*. Theoretical Computer Science, vol. 51, pages 53–80, 1987.

[Wang *et al.* 2015] Zhe Wang, Mahsa Chitsaz, Kewen Wang et Jianfeng Du. *Towards Scalable and Complete Query Explanation with OWL 2 EL Ontologies*. In Proceedings of CIKM, 2015.

[Zhou *et al.* 2012] Liping Zhou, Houkuan Huang, Guilin Qi, Yue Ma, Zhisheng Huang et Youli Qu. *Paraconsistent query answering over DL-Lite ontologies*. Web Intelligence and Agent Systems, vol. 10, no. 1, pages 19–31, 2012.

A

COMPLEXITY OF REASONING IN PROPOSITIONAL LOGIC

A.1 Complexity theory

We study the complexity of different problems, generally *decision problems*, whose answer is yes or no on any input, and a few *generation problems*, which generate a solution (or all solutions). We say that a procedure *solves* a decision problem if it terminates and outputs the correct answer on all inputs, and it solves a generation problem if, for all inputs, it outputs a solution (or all solutions) if one exists, and no otherwise.

Decision problems can be assigned to different complexity classes depending on the computational resources needed to solve them. The following classes of the polynomial hierarchy are relevant to this thesis:

- P: problems which are solvable in polynomial time in the size of the input.
- NP: problems which are solvable in non-deterministic polynomial time.
- coNP: problems whose complement is in NP.
- BH_2: problems that are the intersection of a problem in NP and a problem in coNP.
- Δ_2^p: problems which are solvable in polynomial time with an NP oracle.
- $\Delta_2^p[O(\log \mathrm{n})]$: problems which are solvable in polynomial time with at most logarithmically many calls to an NP oracle.
- Σ_2^p: problems which are solvable in non-deterministic polynomial time with an NP oracle.
- Π_2^p: problems whose complement is in Σ_2^p.

These classes are related as follow: $\mathrm{P} \subseteq \mathrm{NP} \subseteq \Delta_2^p \subseteq \Sigma_2^p$ and $\mathrm{P} \subseteq \mathrm{coNP} \subseteq \Delta_2^p \subseteq \Pi_2^p$, and it is widely believed that all these inclusions are proper.

Importantly, the class $\Delta_2^p[O(\log \mathrm{n})]$ can be equivalently characterized as the class of decision problems which can be solved in polynomial time with a single round of parallel calls to an NP oracle, (cf. [Buss & Hay 1991]). Actually, problems that can be solved by a polynomial-time procedure using NP oracle calls that can be structured as a tree are in $\Delta_2^p[O(\log n)]$ [Gottlob 1995].

The class AC^0 consists of those problems that can be solved by a uniform family of circuits of constant depth and polynomial size, with unbounded-fanin AND and OR gates. It is known that $\mathsf{AC}^0 \subset$ P.

A problem is *hard* for a complexity class if it is at least as difficult as any problem of this class. When showing that a decision problem P is hard for a given complexity class, we use standard polynomial-time many-one reductions (also known as Karp reductions), which transform an instance of one decision problem known to be hard P' into an instance of P. To show that a generation task is hard for a class C, we reduce a C-hard decision problem to it. As we cannot use many-one reductions (which relate two decision problems), we use polynomial-time Turing reductions, that is, we show how to solve the C-hard decision problem using a polynomial-time Turing machine that can use the generation task as an oracle. Moreover, to prove a stronger intractability result, we only allow a single oracle call.

A.2 Propositional logic

We recall here some definitions for propositional satisfiability and related problems. Propositional formulas are defined from a set of propositional variables X, two constants true and false, and a set of logical connectors: $\neg$ (negation), $\vee$ (disjunction), and $\wedge$ (conjunction). Other connectors can be introduced as abbreviations.

Definition A.2.1 (Propositional formulas)**.** Given a set of propositional variables $X = \{x_1, ..., x_n\}$, propositional formulas are defined as follows:

- true and false are propositional formulas
- every $x \in X$ is a propositional formula
- for every propositional formula φ, $\neg\varphi$ is a propositional formula
- for all propositional formulas φ and ψ, $\varphi \vee \psi$ and $\varphi \wedge \psi$ are propositional formulas

We denote by $\mathsf{vars}(\varphi)$ the set of propositional variables that appear in φ. A *literal* is a formula of the form x or $\neg x$ with $x \in X$. A *clause* is a disjunction of literals. A formula φ is in *conjunctive normal form* (CNF) if it is a conjunction of clauses, and in *disjunctive normal form* (DNF) if it is a disjunction of conjunctions of literals. A formula in CNF $\varphi = C_1 \wedge ... \wedge C_k$ is often be seen as the set of clauses $\{C_1, ..., C_k\}$. A formula in CNF is *monotone* if all its literals are of the form x.

The semantics of propositional formulas is given by interpretations. An interpretation, or valuation, ν is a function that assigns to each variable $x \in X$ a truth value in $\{\mathsf{true}, \mathsf{false}\}$. If the domain of ν is a proper subset of X, ν is called *partial*. The truth value of a formula φ in an interpretation ν, denoted $\nu(\varphi)$, is defined as follows:

- $\nu(\mathsf{true}) = \mathsf{true}$ and $\nu(\mathsf{false}) = \mathsf{false}$

- for every propositional formula φ, $\nu(\neg\varphi) = \mathsf{true}$ if $\nu(\varphi) = \mathsf{false}$ and false otherwise
- for all propositional formulas φ and ψ:
  $\nu(\varphi \vee \psi) = \mathsf{true}$ if $\nu(\varphi) = \mathsf{true}$ or $\nu(\psi) = \mathsf{true}$, and false otherwise
  $\nu(\varphi \wedge \psi) = \mathsf{true}$ if $\nu(\varphi) = \mathsf{true}$ and $\nu(\psi) = \mathsf{true}$, and false otherwise

Definition A.2.2 (Satisfiability, unsatisfiability, tautology)**.** An interpretation ν *satisfies* a propositional formula φ if $\nu(\varphi) = \mathsf{true}$. If there exists a interpretation that satisfies φ, then φ is *satisfiable*. Otherwise, φ is *unsatisfiable*. If every interpretation satisfies φ, φ is *valid* and called a *tautology*.

A *model* of φ is the set of variables assigned to true in a valuation that satisfies φ.

If S and H are two sets of soft and hard clauses such that $S \cup H = \{C_1, ..., C_k\}$, a subset $M \subseteq S$ is a *minimal unsatisfiable subset* (MUS) of S w.r.t. H if $M \cup H$ is unsatisfiable, and $M' \cup H$ is satisfiable for every $M' \subsetneq M$. A *minimal correction subset* (MCS) of S w.r.t. H is a subset $M \subseteq S$ such that $(S \backslash M) \cup H$ is satisfiable and $(S \backslash M') \cup H$ is unsatisfiable for every $M' \subsetneq M$. The sets of MUSes and MCSes are hitting set duals of one another: the MUSes are the minimal hitting sets of the MCSes and vice versa. The *maximal satisfiable subsets* (MSSes) are the complements of the MCSes in S.

Definition A.2.3 (Quantified Boolean Formula)**.** A *quantified Boolean formula* (QBF) is a formula in quantified propositional logic where every variable is quantified, using either existential or universal quantifiers, at the beginning of the sentence. The truth value of a QBF is defined recursively:

- $\exists x_1, ..., x_n \varphi(x_1, ..., x_n)$ is true iff $\varphi(x_1, ..., x_n)$ is satisfiable,
- $\forall x_1, ..., x_n \varphi(x_1, ..., x_n)$ is true iff $\nu(\varphi(x_1, ..., x_n)) = \mathsf{true}$ for every valuation ν of $\{x_1, ..., x_n\}$,
- $\exists x_1, ..., x_n \varphi(x_1, ..., x_n, x_{n+1}, ... x_m)$ is true iff there exists a valuation ν of $\{x_1, ..., x_n\}$ such that $\nu(\varphi(x_1, ..., x_n, x_{n+1}, ... x_m))$ is true,
- $\forall x_1, ..., x_n \varphi(x_1, ..., x_n, x_{n+1}, ... x_m)$ is true iff for every valuation ν of $\{x_1, ..., x_n\}$, $\nu(\varphi(x_1, ..., x_n, x_{n+1}, ... x_m))$ is true.

A QBF is *valid* if and only if it is true.

A.3 Problems used in reductions

We list here the problems of propositional logic we use to show hardness by reduction in the complexity proofs of this thesis.

A.3.1 NP or coNP-hard problems

The following problems are NP-complete:

- SAT: decide if a propositional formula is satisfiable. SAT is already NP-complete for formulas in 3-CNF, i.e., conjunctions of three-literal clauses (3SAT). NP-hardness of SAT holds if we impose that at least one variable appears in positive and negative form in the formula (Lemma 5.2.14).
- decide if a clause of a propositional formula in CNF belongs to every MUS is NP-complete [Liberatore 2005].

The following problems are coNP-complete:

- UNSAT: decide if a propositional formula is unsatisfiable. UNSAT is already coNP-complete for formulas in 3-CNF.
- decide if a valuation that satisfies a monotone 2-SAT formula assigns a smallest number of variables to true (coNP-hardness can be shown by a straightforward reduction from the complement of the well-known NP-complete vertex cover problem).
- Tautology: decide if a propositional formula is a tautology (since φ is valid if and only if $\neg\varphi$ is unsatisfiable).
- given a set $\{C_1, ..., C_k, C_{k+1}\}$ of clauses such that $\{C_1, ..., C_k\}$ is satisfiable and C_{k+1} is not a tautology: decide whether $\{C_1, ..., C_k, C_{k+1}\}$ is satisfiable (Lemma 5.2.15).

A.3.2 BH_2-hard problems

The following problems are BH_2-complete:

- SAT-UNSAT: given a pair (φ_1, φ_2) of propositional formulas, decide if φ_1 is satisfiable and φ_2 unsatisfiable.
- given two sets of soft and hard clauses S, H, deciding if $M \subseteq S$ is a MCS of S w.r.t. H (Lemma 5.3.4).
- decide if a set of clauses of a propositional formula in CNF is a MUS [Liberatore 2005].

A.3.3 Δ_2^p or $\Delta_2^p[O(\textbf{log n})]$-hard problems

The following problem is Δ_2^p-complete:

- Lexicographically maximum truth assignment problem [Krentel 1988]: given a satisfiable 3CNF formula $\varphi = C_1 \wedge \ldots \wedge C_k$ over variables $x_1, ..., x_n$, decide whether the lexicographically maximum truth assignment satisfying φ with respect to $(x_1, ..., x_n)$, denoted by $\nu_{\max}$, fulfills $\nu_{\max}(x_n) = \mathsf{true}$.

The following problem is $\Delta_2^p[O(\text{log n})]$-complete:

- Parity(SAT) problem [Wagner 1987, Eiter & Gottlob 1997]: a Parity(SAT) instance is given by a sequence $\varphi_1, \ldots, \varphi_m$ of propositional formulas in CNF, and the problem is to decide whether the number of satisfiable formulas is odd. It is known that it can be assumed w.l.o.g. that the formulas are such that φ_{i+1} is unsatisfiable whenever φ_i is unsatisfiable. Consequently, the problem reduces to deciding existence of an odd integer p such that φ_p is satisfiable and φ_{p+1} is unsatisfiable. $\Delta_2^p[O(\log \text{n})]$-hardness holds if the propositional formulas are in 3-CNF (Parity(3SAT)).

A.3.4 Σ_2^p or Π_2^p-hard problems

The following problems are Σ_2^p-complete:

- $\text{QBF}_{2,\exists}$: decide the validity of a $\text{QBF}_{2,\exists}$ formula

$$\exists x_1, ..., x_n \forall y_1, ..., y_m \varphi(x_1, ..., x_n, y_1, ..., y_m).$$

  Σ_2^p-hardness holds when $\varphi(x_1, ..., x_n, y_1, ..., y_m)$ is a a 2+2 DNF formula, i.e. $\varphi(x_1, ..., x_n, y_1, ..., y_m) = \bigvee_{i=1}^k C_i$, where $C_i = \ell_1^i \wedge \ell_2^i \wedge \neg\ell_3^i \wedge \neg\ell_4^i$.
- deciding if a clause of a propositional formula in CNF belongs to a MUS [Liberatore 2005].

The following problem is Π_2^p-complete:

- $\text{QBF}_{2,\forall}$: decide the validity of a $\text{QBF}_{2,\forall}$ formula

$$\forall x_1, ..., x_n \exists y_1, ..., y_m \varphi(x_1, ..., x_n, y_1, ..., y_m).$$

  Π_2^p-hardness holds when $\varphi(x_1, ..., x_n, y_1, ..., y_m)$ is a 3-CNF formula.

B

RÉSUMÉ EN FRANÇAIS

Interrogation de données en présence d'ontologies

L'accroissement du volume des données disponibles pose le problème de savoir les exploiter. Interroger les données de façon précise et efficace est une tâche complexe. Il est notamment nécessaire de permettre l'intégration de données provenant de différentes sources utilisant des vocabulaires différents et la formulation de requêtes d'une manière simple et intuitive, dans un vocabulaire proche de celui de l'utilisateur. Par exemple, supposons que quelqu'un cherche à trouver les professeurs d'un département qui enseignent un cours lié à l'intelligence artificielle à partir d'une liste des membres du département avec l'intitulé de leur poste et une liste des cours avec leurs enseignants. Si il intègre simplement ces données dans une base de données traditionnelle et formule directement sa requête "sélectionner tous les professeurs qui enseignent l'intelligence artificielle", il peut rencontrer les problèmes suivants: d'abord l'intitulé des postes ne sera pas "professeur" mais plutôt "professeur titulaire", "professeur assistant", ou "professeur associé" par exemple, ensuite de nombreux cours qui concernent un sous-domaine de l'intelligence artificielle ne seront pas pris en compte par cette requête (par exemple les cours répertoriés comme traitant de "raisonnement automatique", "représentation des connaissances" ou "logiques de description" ne seront pas reconnus comme des cours d'intelligence artificielle). Il doit donc d'abord trouver les différentes dénominations qui correspondent à un poste de professeur et les différents domaines et sous-domaines de l'intelligence artificielle, puis reformuler sa requête en conséquence. L'interrogation de données en présence d'ontologies est un paradigme récent qui ajoute une couche sémantique au-dessus des données à l'aide d'une théorie logique appelée ontologie, qui formalise la connaissance à propos d'un domaine d'intérêt et est utilisée pour raisonner sur les données pour fournir des réponses plus complètes aux requêtes. Dans notre exemple, ajouter aux données une ontologie qui donne des informations sur les domaines et sous-domaines de l'informatique et sur l'organisation de l'université permettra à l'utilisateur d'obtenir toutes les réponses pertinentes en posant sa requête de la manière qui lui est naturelle.

Les logiques de description [Baader *et al.* 2003] sont une famille de fragments de la logique du premier ordre qui sont largement utilisées comme langages d'ontologie. Une base de connaissances exprimée en logique de description est constituée d'une ontologie, appelée TBox, qui exprime des connaissances générales et des règles à propos du domaine d'intérêt, et d'un ensemble de données, appelé ABox, qui donne des informations sur des individus spécifiques. Par exemple, la TBox d'une base de connaissances sur le domaine de l'université pourrait indiquer que les cours sont enseignés par des enseignants et suivis par des étudiants, et sa ABox pourrait spécifier qu'un individu nommé Ann donne un cours de base de données qui est suivi par un autre individu Bob.

Enrichir les données avec une ontologie a un prix: cela augmente la complexité algorithmique de la réponse aux requêtes. Le passage à l'échelle étant essentiel pour les applications riches en données, les logiques de description dites légères, qui offrent un bon compromis entre l'expressivité du langage et la complexité des problèmes de raisonnement associés, ont suscité un intérêt croissant. En particulier, la famille DL-Lite [Calvanese *et al.* 2007] a été spécialement conçue pour l'interrogation de données en présence d'ontologies qu'elle permet de réduire, par réécriture de la requête, à l'évaluation standard d'une requête sur une base de donnée. Dans cette thèse, nous adoptons le langage DL-Lite$_\mathcal{R}$ qui est le dialecte de la famille DL-Lite à la base de OWL 2 QL [Motik *et al.* 2012], le profil du standard pour le web sémantique OWL2 pour la réponse aux requêtes.

Gestion des incohérences

Un problème important qui se pose dans le contexte d'interrogation de données en présence d'ontologies est de traiter le cas où les données sont incohérentes avec l'ontologie. En effet, alors que la TBox est généralement de taille limitée et soigneusement déboguée par des experts du domaine, la ABox est typiquement large, sujette à de fréquentes modifications, et peut résulter de l'intégration de différentes sources de données, ce qui rend les erreurs probables. Une base de connaissances incohérente impliquant toute formule logique, poser une requête sur une telle base renvoie toutes les réponses possibles formées à partir des individus de la base. Par exemple, si une base de connaissances indique qu'il est impossible d'être en même temps un professeur titulaire et un professeur assistant et qu'un individu Ann est indiqué comme étant les deux, la base de connaissances permettra de conclure non seulement qu'Ann est un professeur titulaire et assistant, mais aussi qu'Ann est un cours par exemple, ce qui est clairement indésirable.

Il y a deux attitudes possibles dans ce contexte. La première est de restaurer la cohérence, en abandonnant une ou plusieurs assertions portant sur l'ancienneté d'Ann dans notre exemple, mais il peut être impossible de le faire d'une façon satisfaisante. En effet, nous ne savons souvent pas comment réparer les données (Ann est-elle un professeur titulaire ou assistant?), et supprimer toutes les informations impliquées dans des contradictions résulterait souvent en une perte d'information inacceptable. De plus, examiner à la main chaque conflit dans les données pour réparer une grande base de données serait trop coûteux. La seconde option est de décider de vivre avec les incohérences, en essayant d'obtenir des réponses qui ont du sens à partir de données incohérentes. Par exemple, il peut être acceptable de conclure qu'Ann est un professeur, mais pas que c'est un cours. Plusieurs sémantiques tolérantes aux incohérences ont été définies dans ce but. La plus connue est la sémantique AR [Lembo *et al.* 2010], qui fournit les réponses qui sont vraies dans chaque sous-ensemble cohérent maximal des données, appelé réparation. Cette sémantique revient à accepter les réponses qui sont vraies quelque soit le monde possible choisi. Par exemple, elle permet de trouver qu'Ann est un professeur, sans information sur son niveau d'ancienneté. L'inconvénient de cette sémantique est qu'elle est difficile à calculer. En effet, pour des bases de connaissances en DL-Lite, la réponse aux requêtes conjonctives sous la sémantique AR est intractable, même quand la complexité est mesurée uniquement en fonction de la taille des données. Pour surmonter cette difficulté, une approximation de AR, appelée IAR, a été introduite dans [Lembo *et al.* 2010]. Les réponses IAR sont obtenues en interrogeant l'intersection des réparations. Cette sémantique est aussi intéressante pour elle-même car elle retient seulement les réponses les plus sûres, dont les supports ne participent à aucune contradiction. Sous cette sémantique, aucune information à propos d'Ann ne sera obtenue, car les raisons de penser qu'Ann est un professeur prennent toutes les deux part à un conflit, donc ne sont pas complètement fiables. A l'autre bout de l'échelle, la sémantique brave donne toutes les réponses qui sont vraies dans au moins une réparation [Bienvenu & Rosati 2013]. Il peut en effet être important pour certaines applications de ne manquer aucune réponse possible qui a une raison cohérente d'être vraie. Utiliser cette sémantique permet de trouver qu'Ann peut être un professeur titulaire aussi bien qu'elle peut être un professeur assistant.

Contributions

Le but de cette thèse est de développer des méthodes pour gérer en pratique des bases de connaissances incohérentes. En particulier, nous défendons l'idée que la sémantique AR, bien qu'intractable, peut être utilisée en pratique.

Notre première contribution est en effet une approche pour classer les réponses selon qu'elles sont conséquences de la base de connaissances sous sémantique IAR, AR ou brave. Pour la sémantique AR, les sémantiques brave et IAR fournissent des bornes supérieures et inférieures calculables en temps polynomial par rapport à la taille des données, et une traduction du problème en un problème de satisfiabilité propositionnelle nous permet de décider si les réponses braves et non-IAR sont vraies sous sémantique AR.

Au-delà de l'efficacité de la réponse aux requêtes, il est important de pouvoir expliquer les résultats des requêtes sous les sémantiques tolérantes aux incohérences. En effet, un utilisateur peut naturellement se demander pourquoi une réponse appartient à une de ces classes (e.g. pourquoi Ann est-elle indiquée comme un professeur pour la sémantique AR mais pas pour IAR?). C'est pourquoi notre seconde contribution est un cadre pour expliquer les réponses positives et négatives sous les sémantiques AR, IAR et brave (e.g. Ann est probablement un professeur car c'est un professeur assistant ou un professeur titulaire dans tous les mondes possibles, mais aucune de ces deux raisons n'est hors de doute car elles sont contradictoires).

Notre troisième contribution est une approche de réparation partielle des données guidée par les requêtes. En effet, bien que les sémantiques alternatives soient nécessaires pour utiliser des bases de connaissances incohérentes, elles ne dispensent pas d'améliorer la qualité des données. Nous proposons d'exploiter les retours des utilisateurs sur les résultats des requêtes qui sont corrects ou incorrects pour nettoyer les données en nous concentrant sur la partie utile pour l'utilisateur et qu'il connait suffisamment bien pour la réparer (e.g. si l'utilisateur sait qu'Ann est un professeur assistant, nous pouvons supprimer les données qui indiquent que c'est un professeur titulaire, puisqu'elle ne peut pas être les deux à la fois).

La dernière contribution de la thèse est l'étude de variantes des sémantiques AR, IAR et brave obtenues en remplaçant les réparations classiques par des réparations préférées. Cela nous permet de prendre en compte les informations sur la fiabilité des données (e.g. si l'information qu'Ann est un professeur assistant vient d'une source moins fiable que le fait qu'Ann est un professeur titulaire, nous pouvons garder uniquement les réparations qui contiennent ce dernier fait et conclure qu'Ann est un professeur même pour la sémantique IAR).

Pour chacun des sujets abordés, nous analysons la complexité des problèmes liés et proposons des algorithmes pour les résoudre, en exploitant les performances des SAT solveurs modernes pour résoudre en pratique les problèmes durs. Nous avons mis en oeuvre la plupart de ces algorithmes dans notre prototype CQAPri[1] et avons étudié leurs propriétés empiriquement à l'aide d'une base de connaissances incohérente que nous avons construite à partir du benchmark $\text{LUBM}_{20}^{\exists}$.

Organisation de la thèse

La thèse est organisée comme suit:

Chapitre 2 Ce chapitre introduit l'interrogation de données en présence d'ontologies exprimées en logique de description et le langage DL-Lite$_\mathcal{R}$ adopté dans ce travail. Dans la seconde partie du chapitre, nous passons en revue les sémantiques alternatives qui ont été proposées pour gérer des données incohérentes dans ce contexte.

Chapitre 3 Dans ce chapitre, nous présentons les algorithmes implémentés dans notre prototype CQAPri pour répondre aux requêtes sous les sémantiques AR, IAR et brave en DL-Lite$_\mathcal{R}$. Nous décrivons ensuite le cadre expérimental que nous avons construit pour évaluer notre système et les résultats obtenus, et passons rapidement en revu les systèmes et benchmarks existants.

Chapitre 4 Nous abordons dans ce chapitre le problème d'expliquer pourquoi un tuple est ou n'est pas une réponse à une requête sous les sémantiques IAR, AR ou brave. Nous définissons des explications centrées sur les données pour les réponses positives et négatives, étudions leur complexité pour DL-Lite$_\mathcal{R}$, et proposons des algorithmes pour les calculer en exploitant des SAT solveurs. Nous présentons aussi notre implémentation dans CQAPri et les expériences que nous avons effectuées.

Chapitre 5 Ce chapitre aborde la réparation des données guidée par les requêtes pour des bases de connaissances DL-Lite$_\mathcal{R}$ incohérentes. Nous considérons le scénario suivant: un utilisateur reçoit les réponses à des requêtes sous les différentes sémantiques tolérantes aux incohérences et indique que certaines réponses sont fausses tandis que d'autres sont correctes et devraient être impliquées sous une sémantique plus forte. Le but est de trouver un ensemble de modifications de la ABox (suppressions et ajouts), appelé un plan de réparation,

[1] available at www.lri.fr/~bourgaux/CQAPri

qui résout le plus de problèmes possible. Après avoir formalisé ce problème et introduit différentes notions d'optimalité, nous étudions la complexité de raisonnement lié aux plans optimaux et proposons des algorithmes interactifs pour calculer de tels plans. Dans le cas où seules les suppressions sont autorisées, nous proposons un algorithme amélioré et présentons l'implémentation de ses principaux composants dans CQAPri.

Chapitre 6 Dans ce chapitre nous étudions des variantes des sémantiques AR, IAR et brave obtenues en remplaçant la notion de réparation classique par un type de réparation préférée parmi quatre (e.g. les réparations de cardinal maximal, ou basées sur des niveaux de priorité qui distinguent les assertions plus ou moins fiables). Nous analysons la complexité de la réponse aux requêtes sous les sémantiques résultantes et proposons une approche exploitant un encodage SAT pour celles basées sur les niveaux de priorité, dont la complexité est la même que pour la sémantique AR classique. Nous présentons ensuite notre implémentation de ces sémantiques et son évaluation expérimentale.

Chapitre 7 Dans ce chapitre, nous nous positionnons dans un contexte plus général de l'état de l'art et détaillons certains sujets mentionnés dans les chapitres précédents.

Chapitre 8 Ce chapitre résume nos contributions et indique quelques extensions possibles de ce travail.

Annexe A L'annexe fournit les bases de théorie de la complexité et de la logique propositionnelle, et rappelle les définitions des classes de complexité apparaissant dans cette thèse ainsi que les problèmes utilisés dans les preuves de complexité.

INDEX

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Gestion des incohérences pour l'accès aux données en présence d'ontologies

Mots clefs : Logiques de description, Réponse aux requêtes, Gestion de l'incohérence

Résumé : Interroger des bases de connaissances avec des requêtes conjonctives a été une préoccupation majeure de la recherche récente en logique de description. Une question importante qui se pose dans ce contexte est la gestion de données incohérentes avec l'ontologie. En effet, une théorie logique incohérente impliquant toute formule sous la sémantique classique, l'utilisation de sémantiques tolérantes aux incohérences est nécessaire pour obtenir des réponses pertinentes. Le but de cette thèse est de développer des méthodes pour gérer des bases de connaissances incohérentes en utilisant trois sémantiques naturelles (AR, IAR et brave) proposées dans la littérature et qui reposent sur la notion de réparation, définie comme un sous-ensemble maximal des données cohérent avec l'ontologie. Nous utilisons ces trois sémantiques conjointement pour identifier les réponses associées à différents niveaux de confiance. En plus de développer des algorithmes efficaces pour interroger des bases de connaissances DL-Lite incohérentes, nous abordons trois problèmes: (i) l'explication des résultats des requêtes, pour aider l'utilisateur à comprendre pourquoi une réponse est (ou n'est pas) obtenue sous une des trois sémantiques, (ii) la réparation des données guidée par les requêtes, pour améliorer la qualité des données en capitalisant sur les retours des utilisateurs sur les résultats de la requête, et (iii) la définition de variantes des sémantiques à l'aide de réparations préférées pour prendre en compte la fiabilité des données. Pour chacune de ces trois questions, nous développons un cadre formel, analysons la complexité des problèmes de raisonnement associés, et proposons et mettons en oeuvre des algorithmes, qui sont étudiés empiriquement sur un jeu de bases de connaissance DL-Lite incohérentes que nous avons construit. Nos résultats indiquent que même si les problèmes à traiter sont théoriquement durs, ils peuvent souvent être résolus efficacement dans la pratique en utilisant des approximations et des fonctionnalités des SAT solveurs modernes.

Title : Inconsistency Handling in Ontology-Mediated Query Answering

Keywords : Description logics, Query answering, Inconsistency handling

Abstract : The problem of querying description logic knowledge bases using database-style queries (in particular, conjunctive queries) has been a major focus of recent description logic research. An important issue that arises in this context is how to handle the case in which the data is inconsistent with the ontology. Indeed, since in classical logic an inconsistent logical theory implies every formula, inconsistency-tolerant semantics are needed to obtain meaningful answers. This thesis aims to develop methods for dealing with inconsistent description logic knowledge bases using three natural semantics (AR, IAR, and brave) previously proposed in the literature and that rely on the notion of a repair, which is an inclusion-maximal subset of the data consistent with the ontology. In our framework, these three semantics are used conjointly to identify answers with different levels of confidence. In addition to developing efficient algorithms for query answering over inconsistent DL-Lite knowledge bases, we address three problems that should support the adoption of this framework: (i) query result explanation, to help the user to understand why a given answer was (not) obtained under one of the three semantics, (ii) query-driven repairing, to exploit user feedback about errors or omissions in the query results to improve the data quality, and (iii) preferred repair semantics, to take into account the reliability of the data. For each of these three topics, we developed a formal framework, analyzed the complexity of the relevant reasoning problems, and proposed and implemented algorithms, which we empirically studied over an inconsistent DL-Lite benchmark we built. Our results indicate that even if the problems related to dealing with inconsistent DL-Lite knowledge bases are theoretically hard, they can often be solved efficiently in practice by using tractable approximations and features of modern SAT solvers.